Limiting work in progress but liberating progress in work;

Utilising process simulation and theories of social science to understand the value of Kanban for software development.

James Moriarty

Submitted in fulfilment of the requirements of CT507

Masters in Information Technology (MIT)

College of Engineering and Informatics

National University of Ireland, Galway

Course Director / Thesis Supervisor: Dr. Owen Molloy

August 2014

Abstract

The Toyota Production System first utilised just-in-time with Kanban as a method of visualising the work being completed. Inheriting from lean and agile software development philosophies, Kanban emerged for software development. The Kanban approach has few rules yet it is a powerful tool to evolve a software process.

In this study ExtendSim simulation software is used to model a Kanban approach. The ease of model creation and the potential for adaptability are both assessed. In addition, literature from social sciences are examined for evidence to support the benefits Kanban can bring to people in the software development process.

This study succeeded in creating a simulation of the Kanban approach for software development. A Kanban board was simulated to serve as a foundation from which other processes could be modelled accurately through modification or evolution. Lead time and cycle time indicate performance depending on the changes to work-in-progress limits of the various stages.

Evaluation of social science literature provides support for the human benefits associated with the implementation of Kanban. The simulation model provides meaningful data but, lacking real data prevents the model being broadly valid. Kanban and simulation appear to complement one another which is worthy of future study.

Acknowledgements

I would like to thank my thesis supervisor Dr. Owen Molloy for his guidance and assistance throughout the work conducted for this thesis. Upon his recommendation I began researching Kanban and simulation for this thesis and now I have learned a great deal about each and will endeavour to make use of this newfound knowledge in the future.

I am very grateful to Imagine That Inc. for granting me the use of their ExtendSim AT simulation software to assist this research. The use of this simulation software was an excellent experience that added significantly to the completion of this thesis.

Finally, I would like to thank my family and friends, most notably my several housemates during the year, as their presence has contributed significantly to my education and development over the course of the last twelve months.

Table of Contents

List of Figures

List of Appendices

Chapter 1: Introduction

1.1 Introduction

Kanban is a highly adaptive approach to software development that utilises a few key rules to great effect to succeed at instilling a culture of continuous incremental improvement upon the development process it is applied to. Kanban has seen a surge in interest in recent years thanks to literature that has been emerging since the turn of the century. The Kanban approach will be explored in much greater detail in Chapter 2.

Social science is a broad academic discipline that encompasses many different subjects. Two branches of this field of study will receive considerable attention for the purpose of this study, and those will be psychology and also management. The Kanban approach achieves more than merely a quicker production time, so in Chapter 3 theory from both psychology and management will be used to assess the value of the Kanban approach.

Business process modelling and simulation is a highly valuable tool that can be used to model a process so optimum scenarios are revealed for it, or to weigh-up the current process against a new process that is under consideration for implementation. There are expenses and difficulties associated with successfully simulating a process, but the insights and savings that can be gained through not disrupting the real process can prove invaluable. Chapter 4 looks at this topic in closer detail. The remainder of Chapter 1 will provide a brief history before Kanban was applied to software development and also present the foundations for this study.

1.2 A History Prior to the Kanban Approach for Software Development

1.2.1 Lean Manufacturing

Lean manufacturing is a production philosophy that serves as the origins of much of what will be explored throughout this thesis. The Toyota Production System (TPS) developed in Japan from the 1940s through the 1980s was reported on by Womack, Jones, and Roos (1990). The tenets of the TPS contradicted the mass production focus in America and its advantages were seen as not exclusive to the manufacture of cars. Consequently, lean benefited from an accelerated adoption in the 1990s and even resulted in variations been developed from its key philosophies (Näslund, 2008).

1.2.2 Just-in-Time Production

Before Womack *et al.* (1990) reported on lean manufacturing, an article by Sugimori, Kusunoki, and Uchikawa (1977) went into detail on a few elements of the TPS. The just-in-time philosophy revolves around the concept of necessity and having minimum stock to provide the parts at a specific time while avoiding the creation of surplus inventory. By manufacturing sufficient quantities accurately as well as one item at a time, there is an elimination of waste and an ability to be more predictable in allocation of employees. The Kanban approach that David Anderson later applied to software development in the early 21st century is evident in this paper and is seen as supportive of the just-in-time philosophy as well as getting the best out of employees.

1.2.3 The System of Profound Knowledge

Dr. W. Edwards Deming's work in Japan from the 1950s occurred in parallel to the successes of lean manufacturing. In the 1990s, Deming reported on a management philosophy he had developed called the System of Profound Knowledge which sought to change organisations for the benefit of everyone involved. This theory was formed through the combination of four areas of research, namely: appreciation for the system, knowledge of variation, theory of knowledge, and psychology (Deming, 2000). This ideas put forward by Dr. Deming re-emerge in the Kanban approach as will be discussed later. The System of Profound Knowledge theory created a focus upon multiple aspects of the organisational theory that when addressed collectively can derive significant gains and improvements for the organisation as a whole as well as the customers.

1.2.4 Theory of Constraints

The Theory of Constraints (Goldratt & Cox, 1993) placed a focus upon the balancing a systems flow rather than focussing on capacities directly. The propositions of this theory served as the inspiration for much of David Anderson's work in developing a Kanban approach for software development. Attention is paid to the impact of bottlenecks and work-in-process (also commonly referred to as work-in-progress, abbreviated to WIP in either case) on the flow within a process through the measurement of throughput, inventories, and operational costs (Goldratt & Cox, 1993). The former of the three measurements should be increased and the latter two should be decreased. This philosophy is similar to the aforementioned just-in-time methodology that was part of the lean manufacturing approach.

1.2.5 The Agile Manifesto

Fowler and Highsmith (2001) were two members of the Agile Alliance that formed in 2001 and defined the Agile Manifesto which serves as the point from which many approaches to software development became subsumed in the collective concept of agile methodologies. The Manifesto for Agile Software Development proposed the following:

> "We are uncovering better ways of developing software by doing it and helping others do it. We value:
> *Individuals and interactions* over processes and tools.
> *Working software* over comprehensive documentation.
> *Customer collaboration* over contract negotiation.
> *Responding to change* over following a plan.
> That is, while we value the items on the right, we value the items on the left more."
> (Fowler & Highsmith, 2001, p. 35)

The Agile community encourages the adaption of approaches to fit the project at hand. As a consequence, there are many variations to the approaches that initially existed such that new methodologies then become an agile approach in their own right. West, Grant, Gerush and D'Silva (2010) report that Agile is the leading methodology in software development with 35% of respondents citing it as the most reflective of their process. Iterative (21%) and Waterfall (13%) are some way behind.

1.2.6 Lean Software Development

The Agile Manifesto echoes aspects of lean manufacturing and lean software development had existed in the 1990s before been explained in a structured manner by Poppendieck and Poppendieck (2003). Lean software development aligns the lean philosophy into the agile methodology and results in seven principles, as follows:

1. Eliminate waste
2. Amplify learning
3. Decide as late as possible
4. Deliver as fast as possible
5. Empower the team
6. Build integrity in
7. See the whole (Poppendieck & Poppendieck, 2003)

There are 22 tools put forward that are each associated with one of the aforementioned seven principles. The principles and their tools form the basis of much of what will be discussed in Kanban approach later, though Kanban abides by different principles specifically. This still enables the Kanban approach to align with the lean software development philosophy. The principles of lean software development will be revisited when discussing Kanban in more detail.

1.3 Research Purpose

The purpose of this study is to provide a working simulation of a Kanban software development process and to investigate the true benefits to individuals, teams, and organisations through the use of Kanban. Two research questions echo the purpose of this study and they form the basis of this study's orientation. Given that Kanban is does not have many rules it must abide by and is therefore considered to be the most adaptive Agile methodology (Kniberg & Skarin, 2010).

- How easily can the Kanban software development approach be simulated so that the model created could serve as a foundation to model other development processes?

Kanban is cited as having made significant changes to an organisations culture and collaboration of teams (Anderson, 2010).

- Do theories from social science literature support the benefits associated with implementing the Kanban approach?

1.4 Research Methodology

1.4.1 Simulation Software

Imagine That Inc. have granted the use of their ExtendSim AT simulation software package to assist the completion of this research so this software will be utilised to best replicate the Kanban approach to software development. When abiding by the Kanban philosophy in order to model a software development process, it is important to model the actual process being used and then make corrections or improvements from that model (Anderson, 2010). This study has no specific software development process to model so the

various Kanban implementations from relevant literature will be used to inspire a simulation analogous to a general implementation as per the Kanban approach.

1.4.2 Research of Social Science

Research of the field of social science, primarily psychology and management, will be conducted to successfully analyse whether the benefits to individuals, teams, and organisations are supported by research literature from this field of study. The nature of lean software development, and Kanban in particular, means that the employee and their team are valued highly thus there is now a greater consideration of theories from social science that have the potential to be applied in a software development context.

1.5 Research Limitations

There are several limitations to this study. First, Kanban works through modelling a process as it is then improving that process over time. However, this study had no specific software development process to model so a representative Kanban system has being developed through the influences of various leading authors in the field, primarily Anderson (2010) and Kniberg and Skarin (2010). Second, as no specific software development process could be reached, no real-data could be acquired to see whether real inputs generate real outputs.

Chapter 2: The Kanban Approach to Software Development

2.1 Path of the Kanban Approach into Software Development

There is an increase in research into the Kanban approach during the early 21st century thanks to many reported successful implementations highlighting what advantages it can impart upon a software development process. Kanban had previously emerged within the Toyota Production System as a method of pull production alongside the just-in-time (JIT) production process (Sugimori *et al.,* 1977). Kanban is a Japanese word meaning "signal card" which is its function as an indicator of sufficient capacity for another item to be accepted into the manufacturing or development process. Anderson (2010) took inspiration from this philosophy associated with the manufacture of cars and adapted it to the development of software. His decision to do so is justified by the growing interest and an increased adoption by organisations around the world due is some part to its appealing inheritance from both Agile and lean that proves remarkable effective for software development (Anderson & Roock, 2011). Kanban is not really a software development process in itself but is a methodology that can be applied to whatever process is in use and gradually achieve desirable results for organisations as a whole without necessitating a major change initiative or adoption of a plethora of policies (Anderson & Roock, 2011).

Kanban is concerned with evolving the current process incrementally so it can reach a desired state that that is beneficial for the organisation as a whole. Change management is a tricky balancing act, but an implementation of Kanban helps a team move from a position of familiarity at a pace that is not sufficiently disruptive to prompt significant resistance to change. As will be explored in more detail in chapter four, Kanban promotes a clear Theory O approach to change (Beer & Nohria, 2000) which is a "soft" approach with an attentiveness for the culture of the organisation and its human capital. There are five properties associated with the Kanban approach that originate from the lean philosophy. The following five properties have far reaching benefits for teams and organisations:

1. Visualise workflow
2. Limit work-in-progress (WIP)
3. Measure and manage flow
4. Make process policies explicit
5. Use models to recognise improvement opportunities (Anderson, 2010)

2.2 The Kanban Approach and Lean Software Development

A closer look at the lean software development (LSD) principles (Poppendieck & Poppendieck, 2003) reveals 22 thinking tools that appear very similar to the properties of Kanban's philosophy. Visualising workflow is achieved through the creation of a Kanban board (see Figure 1) which is equivalent to value stream mapping of LSD and this visual tool also enables the team to see the waste in their current process. In fact, Kniberg and Skarin (2010) suggest the use of value stream mapping in order to be able to adequately create a Kanban board. The initial Kanban board should have modelled the process as it exists currently. The following is Poppendieck and Poppendieck's (2003) adjusted list of the seven wastes of manufacturing applicable to software development: partially done work, extra processes, extra features, task switching, waiting, motion, and defects. These are all wastes that the Kanban approach can address and the Kanban board is a suitable visual tool to help bring these to attention either directly or most likely indirectly from querying the reason for slow progress at points on the board.

The thinking tools of pull systems and queuing theory contribute to delivering as fast as possible. These are equivalent to the measurement and management of flow as the Kanban approach places a focus upon reducing cycle-time and increasing throughput. Kanban is a pull system that makes every attempt to preserve a steady flow of work rather thus eliminating waste without explicitly targeting the elimination of waste. The thinking tools of LSD that are concerned with the empowerment of teams will be discussed in chapter four. The five properties of Kanban do not directly address the empowering the team but when Kanban is implemented it can have a positive impact on the teams involved in the process.

The unrestrictive use of the word models in the fifth property invites the consideration of theories equally from natural sciences or social sciences in order to develop the Kanban approach further. Goldratt's theory of constraints, Deming's system of profound knowledge, and the theories associated with lean manufacturing and JIT production are all models noted by Anderson (2010) as having been commonly used with Kanban. Chapter 1 has already alluded to each of these and it is clear to see how Kanban has evolved from the collective consideration of these proven models. The flexibility of the fifth property grants permission for organisations to independently adjust their approach as per theories that they feel can improve their process. Consequently, with experimentation and evolution such as this it stands to reason that Kanban in its infancy still has significant benefits to offer in the future.

2.3 Limiting Work-in-Progress (WIP) and Distinction from Scrum

The notion of limiting WIP was influenced from the theory of constraints (Goldratt & Cox, 1993). Clearly defining the boundaries of the process to be modelled is important because within this range the amount of work that can be completed at various stages will be capped by the quantitative WIP limits. There is a caveat that the items passing through the stages of the development process must each be of approximately the same size (Kniberg & Skarin, 2010). Failing to do so will necessitate the Kanban process to be adjusted, for instance, adding horizontal swim lanes to the Kanban board to cater for small, medium, and large items separately. Reducing variability by working on similarly sized items allows the whole process to be more predictable with accurate lead times so project managers can then make achievable promises to customers.

An abridged version of what can be considered as a Kanban approach is suggested by Kniberg and Skarin (2010) where they propose that visualising the workflow, limiting WIP, and measuring lead time are sufficient guidelines for successful implementation. Scrum and Kanban are presented alongside one another in this book thus making it easier to learn the distinctions between the two approaches and providing insight into how well they can complement one another. Various distinctions exist between Scrum and Kanban, for instance, the former commits to a planned number tasks for the upcoming sprint and once started they do not waiver from their selection until the selected work is completed (Kniberg & Skarin, 2010). The contrasting offering from Kanban is based on capacity and continual flow through the system. Priority requests can be assimilated into the process next at the expense of less important requests in the backlog, provided the WIP limits indicate there being sufficient capacity to do so. This adaptability to the changing needs of customers illustrates a strength of choosing the Kanban approach. Furthermore, the work being conducted through the use of a Kanban board is constant because there is a continual pulling of tasks through the system which differs to the resetting of Scrum upon completion of sprints (Kniberg & Skarin, 2010).

Despite the similarities and shared philosophies of Scrum and Kanban a team may not be suited to one approach yet really achieve their goals with the other. An interesting case observed by Terlecka (2012) required the use of a Scrum and Kanban hybrid after the team in question had shown no improvement with the implementation of Scrum first and then failed in an attempt at implementing Kanban. This hybrid approach is quite common now,

especially with Corey Ladas' proposition of Scrumban. The lack of buy-in from a single individual did slow the performance as a whole so this highlights the need for a collaborative approach or difficulties will arise. Mahnic (2014) also addressed the combined approach due to the ability for Scrum and Kanban to complement one another, noting that the use of Scrumban has doubled in the last year. The generic Kanban board proposed is a helpful starting point for teams, but as Mahnic failed to point out, the intention is to model the process as it is and then attempt evolutionary change from there. This paper almost implies that the team can adopt the stages presented in the paper.

2.4 The Kanban Board

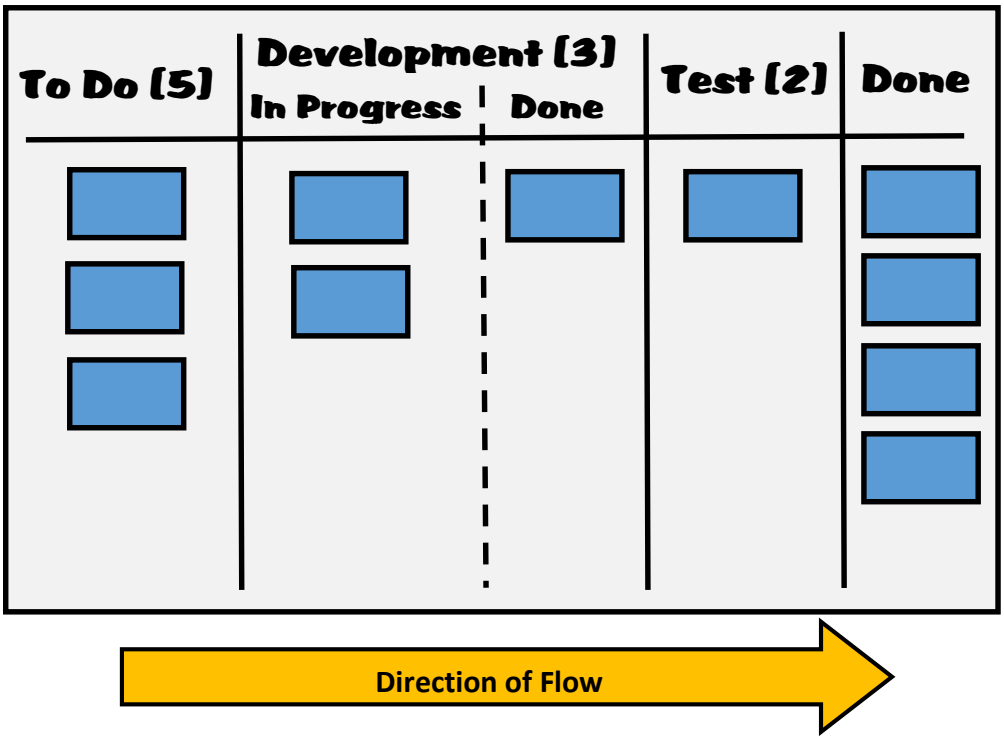


Figure 1: A Kanban board (Adapted from Kniberg & Skarin, 2010, p. 5)

The Kanban board (see Figure 1) has been mentioned already and its use is core to much of what is proposed by this approach to software development. The stages observed in the process are each modelled on this board and WIP limits are decided for each phase. As Kniberg and Skarin (2010) stress, experimentation is what will eventually allow you to understand what limits are most suitable for your organisation's development process. Here is where the addition of simulation software can prove advantageous to an organisation. Modelling an adapted version of the current process in Kanban form would allow managers

to set less arbitrary limits initially so their team can get close to their optimum limits quicker when first adopting the Kanban approach.

The aforementioned WIP limits indicate the maximum number of items that a team can be working on at any one time. This proves highly successful as it eliminates the ability to take on too much work while also increasing the likelihood of higher quality being produced as attention is focussed on fewer items at any one time. The JIT philosophy's pull production was achieved through the use of Kanban as it was within the Toyota Production System (TPS). The current proposition of the Kanban approach retains the action of pulling work through the system which is beneficial because workers should not be overburdened with too many tasks. Managing the flow of items is also easier with the visible presence of a Kanban board because then bottlenecks become apparent as there will be an imbalance of work items before the bottleneck and an absence of them after it. These are all elements that Goldratt's theory of constraints has contributed to the approach.

2.5 Tasks and Bottlenecks

Software development teams have categories of tasks to complete such as features or tech stories and then also the readdressing of bugged versions of either of the aforementioned items (Kniberg, 2011). The Kanban approach is highly suitable to complex environments so on the Kanban board each of these different task types should be indicated on different colours of card. For greater structure, swim lanes can be utilised so that different teams will each be responsible for their own horizontal lanes (Rautiainen, 2010). This is a transparent method of presenting multitasking and it also is advantageous for advancing all categories of task along at a steady pace while being able to observe traditional indicators such as bottlenecks for teams to swarm upon.

The Kanban approach adjusts well when a bottleneck is encountered. Given that the whole system stands to grind to a halt if a bottleneck is not resolved, swarming occurs such that the problematic point is alleviated through the actions of idle team members allocated to other stages (Anderson, 2010). Understandably, for swarming to be effective there must be some level of cross-training or ability for other teams to prove useful at a stage that is not their primary responsibility. This is not prescribed by the Kanban approach, but there is an implied necessity for a degree of flexible skill-sets from the various teams. The nature of the

Kanban approach means that if a bottleneck does arise there is an eventual point at which team members become idle because the WIP limits prevent them from working on more and more items. This relatively simple layout of Kanban has substantial upside given its ability to not find itself caught out by the work of other individuals. The Kanban board and the Kanban approach itself engages the team in a collaborative effort and proactive communication which is reminiscent of the JIT philosophy.

2.6 Case Studies of the Kanban Approach

A blend of quantitative and qualitative analysis was utilised by Sjøberg, Johnsen, and Solberg (2012) to best demonstrate the improvements achieved through the use of the Kanban approach by an organisation that advanced from its previous use of Scrum. Despite positive findings, Sjøberg *et al.* (2012) are quick to caution that this may be a residual effect of implementing Kanban subsequent to the use of Scrum. This is a valid limitation because Kanban can prove highly advantageous once added to software development methodologies. A valuable future study would be in an implementation of Scrum after Kanban to eliminate the concern of causality. Exclusively qualitative investigation has also being used to assess Kanban's suitability to software projects (Ikonen, Pirinen, Fagerholm, Kettunen, and Abrahamsson, 2011) with positive reporting on the autonomous teams abilities to self-organise as well as the value of the Kanban board as a visual tool that manages to improve motivation.

Even hierarchical organisational structure and bureaucratic policies can reap benefits through the use of the Kanban approach. Norrmalm (2012) reports on efforts to increase communication and cross-training of teams. In a context where the process is not working well, such efforts can result in swifter resolution of bottlenecks and no reneging on WIP limits as cross-training allows for swarming. BBC Worldwide (Middleton & Joyce, 2012) almost exclusively used Kanban boards in addition to their Scrum methodology so that they could limit WIP and as a visualisation tool for bottlenecks. This team also reported a process maturation as a by-product of using a Kanban board. Visualising the process and limiting WIP gives a greater clarity that is invaluable to incremental evolution. Improved skills and collaborative efforts were revealed too, similar to those reported by Norrmalm (2012). The Kanban approach does not define the need for these cross-functional teams but the value of doing so is implied as swarming is used for resolution of bottlenecks.

An excellently straightforward and succinct explanation Kanban approach is provided by Reese and Iberle (2011) with the valuable addition of a Hewlett-Packard (HP) case study to illustrate the content. When wishing to implement Kanban it is almost necessary to first educate individuals in the Kanban approach as it has only gained prominence in recent years. HP took time to do so with their staff and also gained employee buy-in through the collaborative creation of the Kanban board. It is clear from this study that the authors and HP understood the essence of the Kanban approach as being larger than a focus on the literal development process. A recent studying surveying organisational performance and employee satisfaction before and after the addition of Kanban boards to their process reported greater project throughput and increasingly positive satisfaction from employees even after eighteen months (Ingason, Gestsson, & Jonasson, 2013). This is quite the endorsement for the Kanban approach but an organisation must not allow itself stagnate so ensuring continuous improvement as Kanban advocates should retain, if not increase, these positive findings over another period equally long.

2.7 Customising the Kanban Board

Some personalisation of the Kanban approach is illustrated by Kniberg (2011) through the assignment of avatars to work items, indicating the responsibility of an individual associated with that avatar as well as presenting a high-level goal and an attainable milestone. These serve to prevent confusion over task assignment and steer the team as a whole in the same direction. These minor additions to the Kanban process are non-prescriptive but hold a value because without such transparency issues could arise that would derail development efforts. The use of swim lanes as well presented by Rautiainen 2010) is another element of the board that can be adjusted to suit the team using it. Depending on the layout of the software development team there can be agreed methods of addressing various categories of tasks such as Kniberg's (2011) use of top lists of features, tech stories, and bugs that can be worked upon in an agreed proportion.

2.8 Bug Maintenance

At the testing stage there is a likelihood of bugs being found in developer's code from time to time. The fact that Kanban chooses to have a steady flow of appropriately sized work

items increases how convenient they are to test and ultimately locate any bugs that may exist. The Kanban approach has an interesting effect on bug maintenance because items are debugged in a steady flow rather than an entire product at once (Kniberg, 2011). Testing time is comparable to doing it all at once at the end but having debugged smaller sized items week by week, there is an increased knowledge of solutions that speeds up fixing time of bugs over time. Returning bugged items to developers as immediately as capacity allows also means that the item is still fresh in their mind so the pace of repair is not as challenging due to relative familiarity.

2.9 Statistics and Metrics for the Kanban Approach

The use of Cumulative Flow Diagrams (CFDs) are prevalent in Kanban literature because they are a strong medium of presenting WIP, lead times, and system issues all in the one chart (Corona & Pani, 2013). Despite the popularity of CFDs, it is interesting that few of the 22 tools created to support Kanban actually include CFDs for presentation of statistical information and furthermore, only five of the tools allow for horizontal swim lanes (Corona & Pani, 2013). Anderson (2010) reports that is the lines are smooth at a relatively universal height respectively then the system is flowing steadily. CFDs can get very messy when looking at data over a lengthy period because data gets bunched in the confines of the graph (Kniberg, 2011). This is a drawback to the use of CFDs but over shorter spaces of time they are an invaluable tool to indicate multiple metrics through a single graphical representation.

Lead time is often offered with varying definitions but the intended use here is that lead time is the time it takes from customer request to customer acceptance testing. Hence, lead time is being considered the time delay that the customer experiences because over this time work is not being conducted exclusively on their product. This is where the concept of cycle time exists within lead time. Cycle time will be considered as the time taken for an item to be selected from the backlog up to the point which it passes testing. This is the amount of time that the item spent in the actual Kanban system and is an invaluable metric for predictability that is necessary for reaching a service-level agreement with customers (Anderson, 2010).

Other metrics of value include due date performance, throughput, flow efficiency, and initial quality (Anderson, 2010). When fixed delivery days are agreed with a customer there

is then a necessity to be measuring whether the team are coming good on their promises. There are specific targets and the actual lead times should be recorded against these as a measure of due date performance. Throughput is the number of items completed and is a value that is desirable to have increasing, provided the process is not failing in measures of quality. Flow efficiency serves to measure lead time of an item against the amount of time it spent at stages that are activity based. The number of bugs recorded per feature is an indication of the quality being produced. Such results can drive a need to improve the process which is very much in line with Kanban's culture of continuous improvement.

2.10 Advantages of the Kanban Approach

Given that the Kanban approach draws from both the philosophy of lean as well as Agile, there is an acceptance from different schools of thought towards its adoption. The increased adoption worldwide was highlighted by Anderson & Roock (2011) and in some part this is down to it being a suitable approach to apply in addition to a process that is already in place. There is also the benefits of a change in culture towards the kaizen ideology whereby there is continual improvement and increased collaboration between team members (Anderson, 2010). The beneficial cultural changes could be just what an organisation needs to really get the best out the process they were employing before the implementation of Kanban. The few rules that govern Kanban means that it is relatively easy to teach to a group and because the use of visualisation is so important, this can be phased in alongside the current process without disrupting the organisation. An incremental adoption of Kanban can be easily achieved through its non-invasive approach.

The flexibility and openness of the Kanban approach to potential improvements is a very strong characteristic because it draws upon theories from a wide range of fields of study that gives it a very well-rounded and considerate approach for all elements of the organisation. The essentials remain the same, but there is scope to modify Kanban if a particular theory resonates strongly with the team. Numerous case studies of Kanban implementations have reported solid findings (Iknonen *et al.*, 2011; *Reese* & Iberle, 2011; Middleton & Joyce, 2012; Sjøberg *et al.*, 2012; Ingason *et al.*, 2013) that strengthen its case for consideration by even more organisations. The use of the Kanban board as a visual tool is extremely effective in providing a clarity and transparency of what work is to be completed and to see the overall picture to assess whether progress is being made or not.

2.11 Disadvantages of the Kanban Approach

Kanban is not actually considered a software development process in its own right. As mentioned earlier it is applied to processes that already exist because the Kanban approach provides incremental evolutionary change to what is already in place (Anderson, 2010). This is an obvious drawback because there is a dependency on other methodologies to be in place before Kanban can then perform its corrective action. This leads into its similarity with Scrum which is made clear by Kniberg and Skarin (2010) as well as the inception of Scrumban by Corey Ladas. These similarities can lead to some confusion over what is actually exclusive to Kanban and as recently as 2013, Corona and Pani report that it still lacks a standard definition within software development despite its steady growth.

With such ambiguities surrounding what is exclusively Kanban, there can be incorrect implementations of Kanban as there is not a true understanding for the value that Kanban provides beyond the actual development process itself. Taking the key features of Kanban without truly understanding what the philosophy of Kanban is would yield results below par because the true benefits of Kanban would not have been realised. Encountering bugs at the testing stage is an understandable likelihood but theoretically in Kanban the flow moves from left to right so in order to remedy a bug the item must move right to left back to the developers which creates additional routing and queuing for re-entry to the development stage due to WIP limits.

Chapter 3: Social Sciences and the Kanban Approach

3.1 Consideration for the Individual

Within the Agile Manifesto (Fowler & Highsmith, 2001) there is the proposal of "individuals and interactions over processes and tools" (p. 35) with the caveat of valuing the latter but valuing the former more. Cockburn and Highsmith (2001) spoke of the need for greater attention on the dynamics of people because the processes suggested by Agile are designed to be flexible which allows a greater degree of discretion from advocates as to what suits them, their employees, the teams, and the organisation the best. An understanding of the knowledge, skills, and abilities (KSAs) of individuals as well as teams allows for greater implementation of agile methodologies as there is a broader consideration factored into the choices (Cockburn & Highsmith, 2001). Laanti (2013) has recently found support for the use of agile methodologies through an examination of changes in stress, empowerment, and performance once Agile had been adopted in the organisation. It is worth noting that Kanban and Scrum did not present significant differences from one another but there is still significant value to teams that choose to use agile methodologies.

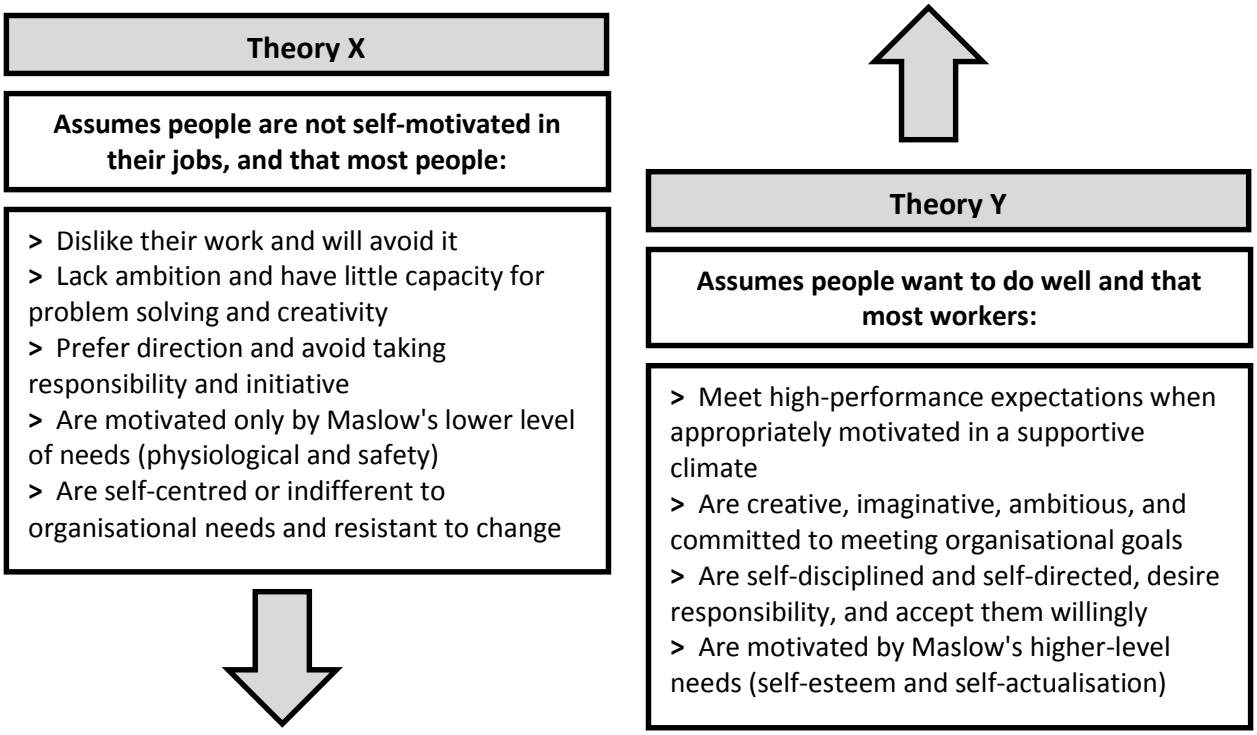


Figure 2: Theory X and Theory Y, McGregor (1960, as cited in Wong, 2007, p. 24)

In 1960, Douglas McGregor proposed Theory X and Theory Y which are contrasting assumptions of how employees are motivated (see Figure 2). Employees lacking self-motivation represent Theory X because they dislike their work, are resistant to change, and would prefer to be directed because they are indifferent to the what the greater need of the organisation (Wong, 2007). By contrast, Theory Y are the employees who want to do well and meet expectations when motivated as well as taken on responsibilities. The objectives of software development require creativity and imagination which are strong indications of a Theory Y disposition. This bodes well for a theory such as Kanban that seeks to strengthen cultural and collaborative aspects of the organisation. There would be significant difficulties encountered if there was a dominance of Theory X in the team and an attempt was made to implement Kanban.

3.2 Managing Individual's Flow

Jim Benson (2012), author of Personal Kanban, made a connection between the works of the psychologist Mihayli Csikszentmihalyi and the Kanban approach. The Kanban approach is concerned with managing flow through the system and there is interesting applications of Csikszentmihalyi's work on psychological flow to the Kanban approach in particular. To successfully promote this concept of flow in an organisation Csikszentmihalyi (2004) notes that it must be embraced by top-management because without their commitment the effort will not succeed, this is also true for the implementation of a Kanban approach. Clear organisational and performance goals are needed for flow to emerge and this refers back to Kniberg's (2011) suggestion of presenting goals clearly on the Kanban board despite not being prescribed. The Kanban board itself does suffice in making individuals aware of their performance goals (items on the board in their stage of responsibility) though the organisational goals only expand as far as the tasks presented on the board.

Feedback either from people or from the work is also necessary to understand one's progress at work (Csikszentmihalyi, 2004) so through the use of a Kanban board there is a reasonable indication of feedback despite being more of an indication of bottleneck emerging which will demand resolution. Feedback through CFDs and other statistical analyses is not sufficient to satisfy the immediacy of feedback proposed by theory. For the final element for achieving flow in an organisation there is once again an emphasis placed on individuals being very much, as per McGregor, Theory Y in their manner. The pairing of challenges with

individuals of suitable abilities is the most important element in achieving flow and this matching can be seen in Figure 3. WIP limits are advantageous to this goal as the challenge, defined by WIP limits, can be adjusted to equate with the skills the individuals possess. As Kanban is an evolutionary process, over time the optimum scenarios will be discovered so organisational flow can be achieved frequently.

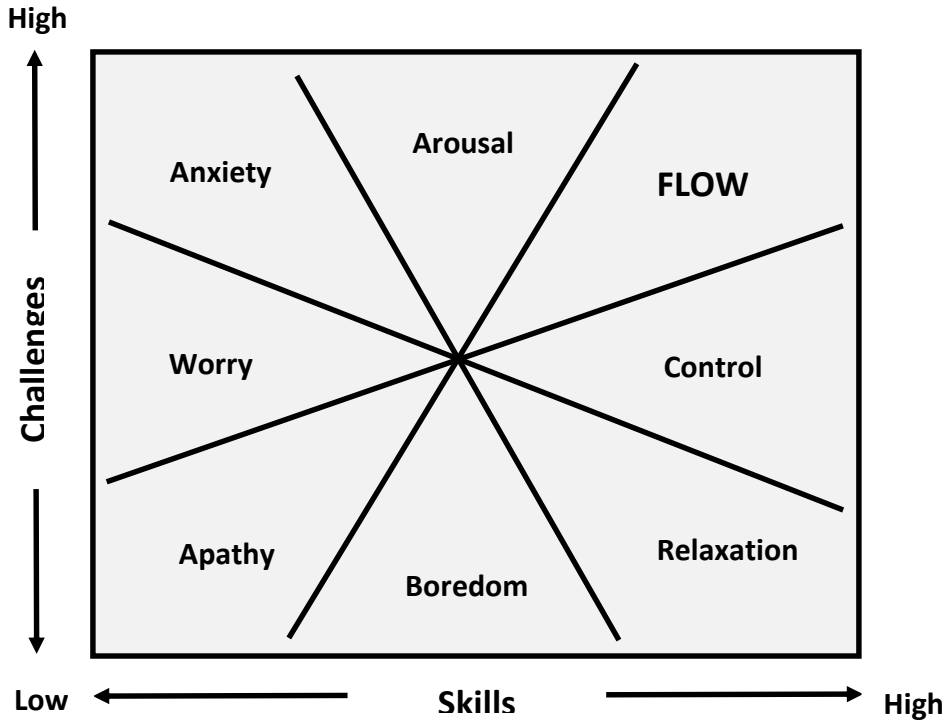


Figure 3: The Map of Everyday Experience (Csikszentmihalyi, 2004, p. 72)

3.3 The Challenge of Context Switching and Incomplete Tasks

Multitasking becomes an issue in workplaces and can be of particular difficult in the software development industry because organisations are catering for multiple projects at any one time and the individuals in the organisation are often required to move from one project to another. Multi-tasking is strongly associated with the capacity of our working memory (Konig, Buhner, & Murling, 2005; Colom, Martinez-Molina, Shih, & Santacreu, 2010) so as increased demands are placed on our working memory, through multitasking, our levels of performance begin to decrease. Weinberg (1992) had observed this effect as context switching (see Figure 4) and this is a strong indication of why it is advantageous for organisations to reduce this impact on their employees. As more projects are juggled by an

employee, they begin to lose significant time on switching between them which leaves little time to actually put the effort into the tasks themselves.

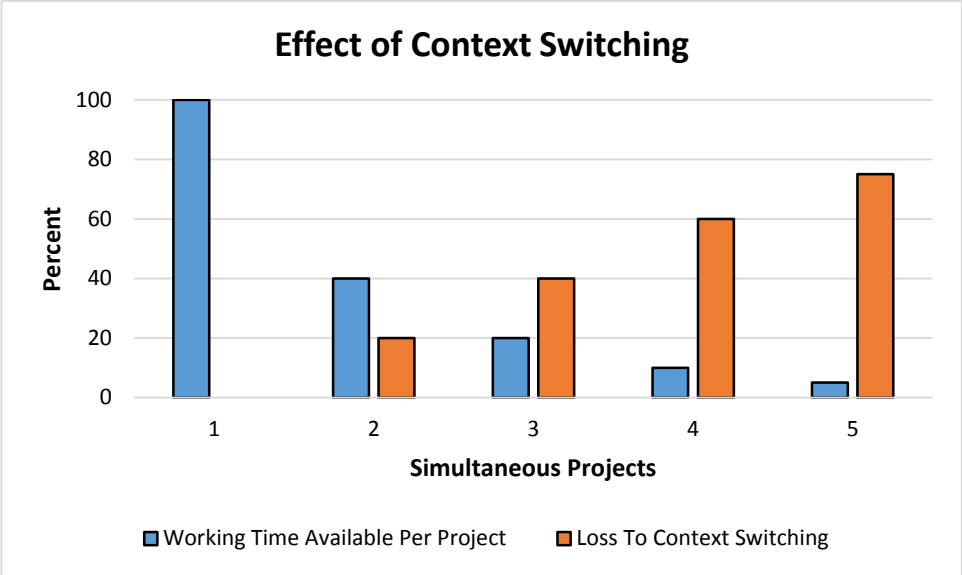


Figure 4: The effect of context switching (Weinberg, 1992)

The Kanban approach in particular serves up a very manageable methodology for solving the issue of multitasking. The frequent collaboration and specification to a particular stage in the development process means that individuals are changing from feature to feature but the overall context remains the same as it is still, for instance, a development or testing duty. The Kanban board has the ability to adapt to get the best out of individuals also as it can be designed to contain horizontal swim lanes which would not only keep employees focussed on their stage in the workflow, but also a specific type of item repeatedly so the context of their work would suffer the least changes. Once items have being selected from the backlog they are committed to being completed so this also avoids there being too much changing of goals beyond the backlog. This and WIP are two key elements that ensure that multitasking does not derail the development efforts of the team once using a Kanban approach in particular.

The persistence of incomplete tasks and goals has been cited by Benson (2012) as one of many cognitive issues that affect individuals working in software development. This is known as the Zeigarnik effect which was initially observed in 1927 and was the subject of recent work by Masicampo and Baumeister (2011) in which they support the use of clearly planned goals to manage the various tasks that are being completed simultaneously. The use of a Kanban board serves as a visual plan that eliminates existential overhead that is abstract

mental models of the multiple tasks that are yet to be completed. The presentation of tasks on the Kanban board alleviates the stress that can manifest through perceived workloads. The true workload may still be significant, but being able to visually perceive the work drives a focus and attention without being overwhelmed by fallacies of the mind (Benson, 2012). The Kanban board has the added advantage of being a shared mental model so that everyone is on the same page.

3.4 The Kanban Approach and Change Management

As Kanban is known to bring about cultural change, especially a kaizen (meaning "good change" though often cited as meaning "continuous improvement") culture. The inheritance from Agile and lean makes this process extremely representative of the Theory O approach as opposed to the more bureaucratic Theory E approach (Beer & Nohria, 2000). The significant encouragement and necessity to experiment (Kniberg & Skarin, 2010) is central to the process of a Theory O organisation as it results in the evolution that Kanban seeks to achieve. David Anderson (2010) works as a consultant and in the case studies he reports his role is very much a supportive and educational one. This is once again indicative of a Theory O organisation as the team are helped to take on Kanban and its associated philosophy they are meant to embrace. The bottom-up participatory element proves important too because Kanban necessitates a responsibility from employees in their roles because the system will stall if one individual is not putting in the work necessary, thus forming a bottleneck. These are aforementioned traits of Theory O are contrary to what is proposed by Theory E so this is a strong indication that Kanban could only prove successful if the organisation already bears some resemblance to Theory O. Kanban could gravitate a Theory E towards the desired state, but the length of time would surely see Kanban removed before it succeeds.

The notion of "good change" or "continuous improvement", through kaizen culture, echoes the idea that organisations must always consider change or they will reach a point at which they find themselves needing to change and their tardiness in realising this means the change initiative will be far more difficult (Vermeulen, Puranam, & Gulati, 2010). The three key points put forward by Vermeulen *et al.* (2010) about fostering communication, building agility, and breaking up entrenched interests, are all aspects that are very much inherent in the Kanban approach to software development. The collaborative nature of Kanban and the

communal role of the Kanban board means the formation of silos is often avoided because there is frequent interaction between the teams responsible for the various stages of the software development lifecycle. Another advantage in adopting a Kanban approach is that there is the encouragement to experiment so there is not one stringent plan that employees must abide by. As Kniberg and Skarin (2010) have remarked, it is as close to doing whatever you want as you could get. The few constraints of Kanban make it function remarkably well but they also allow for the addition of new twists to improve performance. Those that work are held onto to so the process is always in a state of change or "continuous improve" due to the culture it brings forth.

3.5 The Kanban Approach and Team Performance

The influence of content, process, and behaviour on team performance is essential in project teams and these three elements are captured in detail by Wong (2007). Clarity of goals and direction form a key part of content but there must also be a clarity over the process being implemented highlighting the need for training as HP took the time to do with their adoption of Kanban (Reese and Iberle, 2011). The key team behaviours include "mutual trust, interdependency, accountability, valuing individual differences, transparency, learning and recognition" whereas the key team processes are "team meetings, roles and responsibilities, communications, decision making, measuring performance, and team feedback" (Wong, 2007, p. 88). The Kanban approach is certainly effective in delivering on the behaviours and processes that are necessary for project teams to perform well. These are traits that are evident in many teams but the Kanban approach manages to instil these key aspects through a very liberal approach to software development.

Communication is a core element to the effectiveness of all teams, particularly within work teams. Communication can either be downward, upward, or horizontal in organisations that have hierarchies present. Through the use of Kanban boards there is the opportunity for a greater degree of horizontal communication as managers do not need to micro-manage individuals as well as the Kanban boards becoming an indication of progress so significant effort in upward and downward communication can be saved. Managers can choose to intervene when they perceive there to be something amiss and the nature of the Kanban approach means they can engage in a more collaborative approach to goals than formalised communication channels synonymous with hierarchical organisations.

3.6 The Kanban Approach and Leadership

The effectiveness of teams is often dependent on leadership. In order for goals to be successfully achieved there needs to be a clarity in goals and direction so this is where a leader can prove invaluable, even in a Kanban approach that has a high degree of self-managed work. Even though the Kanban board allows for communal understanding of the "big picture" there is still a need for a leader to align the team members as well as motivating them to achieve the objectives they are tasked with. Often the leader would have control over the degree of autonomy that will be allowed for a team but in the case of the Kanban approach the overall philosophy already encourages an autonomous culture with minimal rigor that would usually be presented by management levels. Autonomy at a team level can have differing effects to autonomy at an individual level. Langfred (2005) and Kauffeld (2006) assessed this and found that teams on tasks requiring interdependent teams performed better with high team autonomy but individual autonomy had an adverse effect on their performance. There is further support of such finding by Hoch, Pearse, and Welzel (2010) though they note that age diversity and co-ordination as influences on such performance rather than the task being the key factor.

Decision taking by the supervisor ↘ Decision taking by the employee						
1	2	3	4	5	6	7
Superior decides alone, without consultation of the employees	Superior decides alone, but he tries to convince the employees of the rightness of his decision	Superior decides alone, but he encourages questions concerning his decision mainly to achieve acceptance	Superior informs employees about intended decisions; employees can give their opinion before the superior takes the decision	Employees / group develop(s) proposals. The superior chooses his preferred alternative	Employees / group decide(s); after the superior has shown the goals and problems, and has set up the boundary of the decision authority.	Employee / group decides. The superior serves only as a coordinator to the inside and, mainly, the outside.
authoritarian	**patriarchal**	**informing**	**consulting**	**cooperative**	**delegatory**	**autonomous**

Figure 5. The leadership continuum of Tannenbaum and Schmidt (1973)

The continuum of leadership behaviour proposed by Tannenbaum and Schmidt (1973) (see Figure 5) developed upon the works of Lewin who had proposed the two distinct leadership styles of autocratic versus democratic back in 1948. The degree to which leaders want to

include their employees in the decision-making process can be selected from across this continuum. It is very clear the use of the Kanban approach resides in the delegatory and autonomous bracket with the highest degree of decision making taken by employees. Given that the Kanban approach is often implemented as it possesses the ability to evolve an organisation incrementally over time, this continuum also provides an insight into how difficult the task of implementation would be if the organisation currently employs a deep rooted leadership style representative of the leftmost side.

3.7 Is the Kanban Approach Justified by the Social Sciences?

The Kanban approach is justified in its propositions of improving organisational culture and being a desirable approach for employees. Being anchored in the Agile Manifesto, which espouses a prioritisation for individuals over processes, benefits Kanban because the process improvements emerge once the individuals feel fulfilment in their role. Theory X and Theory Y also indicate the dispositions that would react most favourably with the Kanban approach as Theory X would be quite resistance to change so an attempt of incremental evolution would certainly fail. Theory E and Theory O (Beer & Nohria, 2000) are also supportive of the Kanban approach as it utilises a "soft" approach to change which is attentive to the needs of the individuals and brings change in a bottom-up implementation.

The Kanban approach has strong ties with change management as it is not a specific software process methodology and is suited to changing a process that is already in place. Communication, agility, and breaking up entrenched interests combine to enable continuous improvement (Vermeulen *et al.*, 2010) which are all aspects that the Kanban approach does allow for in abundance. Flow is a key element to the Kanban approach and Csikszentmihalyi's (2004) work on flow in life as well as work carries a value that aligns it with the kaizen culture that is synonymous with Kanban. Limiting WIP hones in attentiveness on specific goals which also prompts an absorbance into a state of flow which is analogous to the activity of the Kanban board.

The Kanban approach's ability to marginalise the significance of multitasking is extremely effective. Keeping individuals in roles where the context of the work remains relatively fixed has major benefits so Kanban is certainly strengthened by how little it concedes to the adverse effects of multitasking. The Kanban board with its WIP limits makes

it clear what tasks are to be taken care of without getting overwhelmed. The Zeigarnik effect and the concept of existential overhead (Masicampo & Baumeister, 2011; Benson, 2012) are both alleviated by the presence of the Kanban board so it is an incredibly advantageous visual tool.

The list of six behaviours and six processes for successful project teams (Wong, 2007) all exist in the Kanban approach. In fact, without any one of these elements the Kanban approach would cease to achieve its kaizen culture and it would stagnate. Autonomous teams (Langfred, 2005; Kauffeld, 2006; Hoch *et al*., 2010) are strongly associated with the Kanban approach and the advantages that this brings about are very evident in management literature. Tannenbaum and Schmidt (1973) also offer a range of leadership styles with Kanban certainly being on the liberal end which is very positive for employees.

All of these theories are renowned amongst literature in the social sciences. The Kanban approach certainly does bring significant benefits that either counter sources of poor performance or encourage sources of high performance. The consideration that Kanban has for individuals, teams, and collaboration is quite apparent and it fits favourably alongside the social sciences, thus confirming the research question into whether the literature of social science support the benefits to be found by using the Kanban approach.

Chapter 4: Simulation Modelling

4.1 Business Process Simulation Modelling (BPSM)

Each year Gartner provides a hype curve which indicates what trends are being hyped up in the various fields of technology. Simulation and optimisation have highly beneficial application in the fields of business process management and in 2011 was approaching its peak of inflated expectations (Dixon & Jones, 2011). At the time it was estimated that it would be two to five years until mainstream adoption of these tools and that range incorporates this year so their use should be very prevalent. However, there are no studies published recently to confirm or deny any mainstream adoption as of yet. The ability for these software packages to give organisations an insight into impact of changes, in a relatively inexpensive way, make them highly valuable to organisations whether they realise this yet or not.

Simulation models are built to serve as imitations of a business process and then these models can be subjected to "what-if" analyses. These simulation models are created through the use of mathematics and logic to relate various elements to one another. Simulation software packages are provided commercially be companies and these almost eliminate the need to be able to code these simulation models as they use predefined blocks that must be connected to one another with some added definitions required in clearly labelled dialog boxes. Simulation software is a very effective tool for prediction of outcomes if a certain method is chosen to be implemented in the organisation or if the value of an alternative design is sought (Laguna & Marklund, 2005).

4.2 Types of Simulation Model

Depending of the content of a simulation model, it can be considered as either static or dynamic, deterministic or stochastic, and continuous or discrete (Laguna & Marklund, 2005). The latter option in each of these cases would be frequently associated with business process modelling and collectively represented through the use of discrete-event simulation (DES). The dynamic element indicates a changing over time whereas the stochastic contribution is through random variables being utilised so specific inputs do not result in specific outputs. The individual units within the system are of importance due to the discrete

nature of this simulation. Each type of simulation model has advantages and disadvantages but each are suitable for different scenarios as demanded. Hybrid simulation models are another option whereby there is the chance combine contrasting model types and use them both to achieve the necessary simulation goals that would otherwise be difficult to achieve through the use of just one of the model types (Kellner, Madachy and Raffo, 1999).

In DES the state of the system is changed by events that occur during the evolution of the system. State variables, such as queues or resource availability, change over time as a result of the events that occur (Laguna & Marklund, 2005). The system is examined from one event to the next because there is no change in the system between events, unlike what would be found in continuous simulation. Queuing systems in particular are simulated well through the use of DES so the use of a Kanban board and having items in queues is suitably equivalent to model Kanban through DES. Lunesu (2013) even cites the suitability of DES for modelling WIP as well as mentioning its suitability as a tool for "what-if" analyses as the outputs of various differing inputs can be compared.

4.3 Advantages of BPSM

Banks *et al.* (2004) and Lunesu (2013) list many advantages of simulation and these significantly outweigh the few disadvantages of simulation that are also offered. Being able to simulate for the outcomes of a business process without committing any resources is extremely cost-efficient especially as the outlay on simulation software is so cheap by comparison. Furthermore, these analyses are being conducted without any disruption to the real business process. When the choice is made to simulate a business process there is a great deal of analysis must be conducted in order to fully understand how the process is operating so it can modelled accurately. This alone is an advantage because it provides an insight into how the process currently runs as opposed to how the process is expected or thought to run. Animated simulations provide a visual insight of the system that also helps in gaining a true understanding of how the process operates.

BPSM is adept at performing "what-if" analyses which are very important in the design of new processes or when changing the current process in favour of another is under consideration. The revealing of how significantly variables impact the system is achievable through BPSM because adjustments in these variables could result in significant gains or

losses in the output. This is very important knowledge to gain as these variables can become the focus of attention in the real business process. It is also noted by Laguna and Marklund (2005) that optimisation technology is now commonly paired within simulation software packages which can bring about significant gains through their use.

4.4 Disadvantages of BPSM

There are a few disadvantages to the use of BPSM but these pale in comparison to the numerous advantages that have been presented already. Four identical disadvantages have been listed by Banks *et al.* (2005) and Lunesu (2013) which suggests that there have not being any newly presented disadvantages in that time that have impacted mainstream adoption adversely. It is costly to purchase commercial simulation software as well as there being a cost in analysing the process, subject of the modelling, sufficiently. However, these costs are significantly less than those that would be incurred through blind application of new plans or through emergency action after having not chosen to change. There is a time consuming element to the analyses and development of the simulation model but these could be weighed up against the cost of implementing change without such preparation.

Creating simulation models requires specialist training, and often experience too, so this can be an additional cost if outsourced or if an employee must be trained. In the case of the latter there would still be an absence of experience though over time this should prove to be a worthy investment. It follows from this that there can be difficulty in interpreting results from these models so once again there would be a necessity for a specialist or knowledgeable individual to reap the benefits associated with simulation. Analytical solutions can sometimes be an option and if ignored in favour of simulation and modelling this is use of such software that was not necessary.

4.5 Software Process Simulation Modelling (SPSM)

Simulation modelling literature has been shown to have increased significantly in quantity during the early 21st century (Jahangirian, Eldabi, Naseer, Stergioulas, & Young, 2010) with interesting additions of added consideration of organisations for with regard to management of projects, knowledge, and training. This evolution of simulation into a broader consideration than merely operations explains the increase in hybrid simulation

modelling which combines simulation techniques to get the associated benefits of each. Some of the prominent disadvantages of BPSM have already being listed (Banks *et al.*, 2005; Lunesu, 2013) but Van der Aalst, Nakatumba, Rozinat, and Russell (2008) have also weighed in with some detailed accounts of the drawbacks associated with current use of simulation. Exploration of design over decision making, lack of reuse of models, and the variability of people and context are not sufficiently considered when creating models (Van der Aalst *et al*., 2008) so this leads to time being wasted on invalid modelling. It is clear that there needs to be improved education of modelling when undertaking the role of modelling a business process or else the choice could be made to outsource to skilled professionals.

Guidance on how to model a software development process was presented by Rus, Neu, and Munch (2003) which goes through the steps of defining goals, questions, metrics and usage scenarios. The information gained from this previous step allows for the sample cases to be created and tested in order to validate the model. An adaptable model resulted that could be used in other scenarios. The work of Rus *et al.* (2003) was developed using Extend V5 which is an older version of the ExtendSim 9 that is utilised in this present study.

4.6 The Kanban Approach and SPSM

The value of having WIP limits in place or having no limits was examined through the development of an object-orientated simulation of the Kanban approach Anderson, Concas, Lunesu and Marchesi (2011). The stages that features go through on the Kanban board each have team members of varying skills simulated to complete the activity. The findings strengthen the case of Kanban having WIP limits in place as well as succeeding in demonstrating the value of Kanban. Kanban is also a strong approach to simulate because the WIP limits allow for adjustment to attain optimised values.

The performance of the Waterfall, Scrum and Lean-Kanban process methodologies has been examined through a system dynamics (SD) simulation (Cocco, Mannaro, Concas and Marchesi, 2011). Despite discrete-event simulation (DES) being the most suitable choice for simulating Lean-Kanban, SD was chosen because DES is not appropriate for the simulation of the other processes being assessed. Each of the processes are implemented differently to be representative of their respective origins. Separate models are built in order to remain true to the distinctions of each process. The less prescriptive nature of Lean-

Kanban with frequent releases and manageable chunks result in effective addressing of errors and the conclusion that Lean-Kanban is the more efficient approach.

Other examinations into modelling Kanban (Cassettari, Gallo, Montella, Revetria, Romano, & Testa, 2012; Turner, Madachy, Ingold, & Lane, 2012) have yielded positive findings in their respective uses for the Kanban approach. The work of individuals in the WIP-limited scenario was evaluated by Turner *et al.* (2012) through Agent-based simulation as suitable for modelling employees as resources. They also simulated the process with an overarching application of statistics in the case of modelling Kanban through DES and continuous modelling. Despite its simplicity Kanban can become more difficult to model as you pay heed to more specific elements within its approach, thus stressing the value of employing hybrid simulation. The former study by Cassettari *et al.* (2012) showed the strategic worth of Kanban as a model of the approach was found to be effective for decision making. The Kanban approach served well in ensuring that the manufacturing company in question had a continuous flow of stock as the recycling of Kanban cards would keep the flow moving and this could be replicated in the simulation model.

The value of WIP-limited approaches to software development and software maintenance were presented by Lunesu (2013) through a simulation of a Lean-Kanban methodology. The study extended as far as being able to model Scrum through slight variations in the model as well as achieving a blend of both to replicate Corey Ladas' proposition of Scrumban. Such a flexible model is highly advantageous for the variations that different organisations would require of it, as well as being providing managers the chance to compare Scrum and Lean-Kanban methodologies to assess which would achieve the most benefit for the development process. The strength of the models proposed is evident in their ability to achieve the same outputs as a real environment through two case studies that the simulation model was applied to.

4.7 A Summary of Simulation Modelling

The hype that was building for simulation and optimisation three years ago has not yet resulted in mainstream adoption as anticipated, or at least the research being conducted into its use has not yet being published. The use of simulation in an organisational environment is highly advantageous though there are some drawbacks that could often be perceived as too

much effort to make it worth investing in, thus limiting the swift embracing by organisations. Kanban, and software development as a whole, are beginning to emerge strongly in the field of process simulation and modelling. Kanban is relatively simple to implement in an organisational setting so this ease carries over to making the simulation of Kanban an achievable task because there are specific phases items go through in a discrete-event manner. Several studies have already being conducted in this regard and each has succeeded in various ways which highlights the absence of any major complexity as a barrier to successfully adopting the Kanban approach and then simulating it in order to broaden the benefits to be reaped from it.

Chapter 5: Simulation of the Kanban Approach

5.1 The Kanban Board

ExtendSim was used to create a simulation model of the Kanban approach as guided by the literature on Kanban and through the assistance of ExtendSim's accompanying user manual. The model presented in Appendix A shows the simulation at rest, albeit with data remaining from a previous run. Through careful consideration it was decided upon to model five overarching stages on the Kanban board, namely: Next, Analysis, Development, Testing, and Complete. A backlog of requests (see Appendix C) does also exist but it is not of any advantage to attempt to represent this on the simulated Kanban board. There is also an independent portion of the Testing stage dedicated to Bugs as these will have to return to the Development stage so they must await sufficient WIP to make the transition back to the previous phase.

Average, minimum, and maximum measures of both cycle time and lead time are visible to the top right of the Kanban board. A distinction has been made between these for the purpose of this study and is familiar in some other studies and literature too. Lead time is being termed the time taken from the Requests block (see Appendix C) to the Lead Time block of Appendix G. Cycle time is being recorded from the Initialise block in Appendix D as far as the Cycle Time block in Appendix G. Additionally there is a shift feature labelled as the Weekly and Daily blocks which is modelling a 40 hour working week so that results are equivalent to what would be achieved each week. The clock in the top left is a necessary Executive Function which effectively ensures the simulation runs.

WIP limits are presented at each stage of development process and these can be manually adjusted from the 'Kanban board' and the change takes effect further down in the system thanks to cloning. An overall WIP limit can be observed beneath the Shift blocks and its value is equal to that of the stages totalled (counting infinity as a zero value for this purpose). This overall WIP is the total number of Kanban cards that are available to the system and they are released from the bottom right of the Kanban board upon completion then recycled to the start again. The final point on Appendix A is the CFD hierarchical block, beneath the Executive Function, which brings up the representation seen in Appendix H that will be explained later.

5.2 The Active Simulation of the Kanban Approach

Appendix B only adds a little additional value to Appendix A as it captures the simulation in motion rather than in an idle state. The Kanban cards that can be seen on the board are not items that are provided by ExtendSim, though there is the provision of numerous colours of spheres. ExtendSim does provide the opportunity to create custom animations through saving images in a choice of a few formats and then into a specific folder. Consequently, this yielded the opportunity to create some Kanban cards rather than having spheres on it. Green cards are seen at the start when selected to the Next column and also when they reach the Complete stage.

Yellow cards are features and blue cards are tech stories as had been addressed by Skarin (2011). Red cards emerge when a bug is found in feature of tech story and these return to the Development stage when there is sufficient capacity. Lastly, an orange card can be seen in the bottom right corner which is representing the Kanban card that has been freed as there is an item that has reached the Complete stage.

5.3 Inside the Hierarchy of the Kanban Board

Appendices C to G represent the direction of flow through the system, with the exception of any bugs that are discovered. In Appendix C, requests get paired with a Kanban card when there is one available from the Kanban block. The Input Queue is what is the queue being represented by the cards in the Next stage on the Kanban board. Various attributes are applied here and throughout the model but they are self-explanatory from the title on the block so there is no need to over indulge in the description of such aspects.

The capacity throughout the system is restricted by the number of Kanban cards available, but at each phase of the development process there are gates that cannot open unless there is sufficient capacity in the area between its sensors. Until an item leaves the Analysed queue in Appendix D, no additional work can be taken through.

The Development stage in Appendix E is relatively simple in its content but the routing at the start of this stage is choosing items based on priority. As items that were found to have bugs return to the developers, there is a priority placed upon their resolution so they are tended to before pulling more items from the Analysis stage.

The most complex of all the hierarchies is the Testing stage (see Appendix F) as there is significant routing and more than one exit from this stage of the process. Items that had already been filed as a bug and then fixed are routed through the top of the ItemType block so that they are not being repeatedly looped through bug fixing, although there is the change for debugged items to have gained a different bug. Features and tech stories that have not already been fixed for bugs stand the chance of being sent to the Bug Queue to await capacity for developers to debug them, or alternatively they will progress on to the Complete stage in Appendix G.

At this final stage there are multiple blocks gathering historical data or values of particular metrics. There is also a block that removes the Kanban card from the item it was hosting and the card returns to the start of the lifecycle to be reused in limiting WIP. The item just awaits release and is then deployed. In Appendices C and G there are values being "thrown" from various blocks. These values are then "caught" within the CFD hierarchy (see Appendix H) and the values combine to present a graph of completed items, item backlog, and WIP. The graphical representation in Appendix H would not ordinarily be expected as there is an expectation for Kanban to be a continuous process that would maintain reasonably steady values over time. The nature of the simulation modelled was that there was a maximum number of 50 items being produced so there was an eventual termination to the process and a finality to the lines on the graph.

Chapter 6: Discussion of the Kanban Simulation Model

6.1 The ExtendSim Experience

Given the inaccessibility to real data from a software development process, or even a software development process to model, this study became more of an exploration of the capabilities of commercial simulation software packages, ExtendSim in this case, than a pursuit to create a validated Kanban approach to software development.

ExtendSim proved very user friendly and the accompanying documentation was extensive but easy to follow, especially when progress was approached in an incremental fashion rather than attempting to quickly reach more difficult aspects of simulation offered by the software. The libraries of blocks allow for a great range of creativity to be employed in order to successfully replicate a process as desired. There is minimal effort to the simulation process thanks to the predefined blocks which means that organisations should give serious thought to the training of individuals in simulation modelling as an employee of expertise with such a tool could yield significant return of investment if they are appropriately trained.

6.2 Suitability of the Simulating the Kanban Approach

The model provided is not exclusive to any one system, it is also likely to not currently imitate any system precisely, unless by sheer chance. However, this model serves as an indication of simulation software packages' ability to create a user friendly simulation in a user friendly way. The three key tenets of Kanban are easily replicated through the use of simulation software. Appendix A presents each of the three aspects; limiting WIP, visualising workflow and measuring lead time (Kniberg & Skarin, 2010). These are all clearly visible in Appendix B so once satisfied with the processing time of items there arises the opportunity to keep running models by adjusting the various WIP limits and seeing how lead time and cycle time change accordingly.

The CFD block is indicative of how anything in the system could be graphed as per the wishes of stakeholders that may demand feedback on a particular aspect of the system. Simulation models are full of statistics at each phase of the process so if a team wishes to see idle time of a specific queue then that can be modelled. This goes to show how complementary the Kanban approach and simulation software are of one another. Despite its

relatively straightforward approach, Kanban does manage to derive key data that drives its improvement culture. Simulation modelling software has the capability to give more statistics than the Kanban approach could ever want so a combination of these two tools in an organisation would serve to exploit every opportunity for improvement.

The addition of swim lanes to one of these simulated Kanban boards is not difficult as is evident from the Bugs section that is visible at the bottom of the Testing column in both Appendix A and Appendix B. Careful use of routing could would enable several paths to completion depending on various criteria that could be applied to the model. The primary difficulty around modelling a Kanban approach is that limiting WIP can lead problems as there needs to be sufficient capacity for the system to be able to pull items through. In the case of rework there needs to be careful attention to WIP limits as a bugged item must have somewhere to reside while remaining in line with the WIP limits of the system. Queues create idle time so these are considered waste which should be limited. Consequently queues cannot just be placed on the model to alleviate problems as this adversely impacts on lead time.

6.3 The Combined Effect of the Kanban Approach and Simulation

If deciding to commence use of the Kanban approach, the combination of the approach and the utilisation of simulation software can prove to be highly advantageous. By adopting Kanban you are forced to think about the process currently in use and similarly in order to simulate a process you must also think about the current process in great detail. Both of these seemingly distant software process tools achieve a complementary goal through a slightly different outlook upon the system. The combination of the two could serve to be particularly potent.

Simulation software offers the option to build from the bottom up or the top down. In the case of the model presented with this study, the model was built from the bottom up but the desired state of the hierarchical structure of the Kanban board's stages were a known but flexible target. If considering the Kanban approach as well as simulation then the flexibility of each can blend nicely to assist one another in reaching the target of an accurately modelled view of the system. The use of hierarchical blocks, as achieved in this model, can group together a set of processes that will then form one stage on a Kanban board. This may either

be identified through a collection of processes' suitability to a particular stage or the commonalities of several processes may drive the consideration of re-evaluating the structure of the Kanban board.

There could understandably be scenarios where a desire for a change in either the Kanban board or the simulation software would necessitate major disruptive change in the other. However, this would prove to be advantageous because such a contrast would not allow for ignorance of limitations that would otherwise go unaddressed on account of being too much hassle. These two differing viewpoints on the one process would serve to produce a far more comprehensive model of the system. The concept of approaching a process with simulation and Kanban independently has not being explicitly stated in previous literature and it serves quite a strong purpose in marrying bottom-up and top-down creation of software process models.

6.4 The Adaptability of the Kanban Simulation Model

The model presented in this study is far from rigid in its design. The possibilities with such a foundation are endless. It would all be dependent on the process being modelled, but any number of stages could be represented on the Kanban board and differing board designs such as the use of swim lanes can easily be attributed with relative ease. Additional degrees of detail can be considered by choosing to have multiple hierarchical levels such that, for instance, within the development stage would exist a Kanban board of stages and items, exclusive to the development team, that are a decomposition of the main Kanban board's items. Herein lies the capability to have the various tiers of the organisation working through the medium of Kanban board and thus creating a more predictable environment overall.

Kanban chooses to break down its work into similarly sized pieces so that the WIP limits work as effectively as they are designed to. Another alternative is to have work categorised into as few groups as possible such that small items would all be grouped together. Through this method various swim lanes could be used to segregate the different categories of work items. Whichever of these approaches are chosen, a real software development process should have a knowledge of a duration range for which items take so this can be easily replicated in the assignment of processing time to work items. As it stands,

the model is not choosing processing time based on real data but the system remains representative of what can be achieved.

6.5 "What-If?" Analyses and the Kanban Simulation Model

"What-if?" analyses can still be conducted on this model though their validity remains in doubt. The various WIP limits have been cloned to the forefront of the system so that adjustments at various points can see a specific change in output. A key element of the entire model is the Mean & Variance blocks which are the two lowest blocks on Appendix G. Through the alteration of the number of runs to a much higher value than just one, the mean value over all these runs is produced as well as confidence intervals which can strengthen the decision making process for an adjustment of WIP limits. Using seed values to get a more regular random generation of values means that changes in results will be purely down to alterations by the modeller as no alteration would have produced the same result because the seed value steers the random generation of values.

6.5 Resource Allocation for the Kanban Simulation Model

Resources are an additional facility that were initially included but then omitted because the model already had a high degree of hypothetical elements without adding labour pools also. This did work well when implemented though because the Resource Manager block allows for optimum allocations of resources as well as the specification of multi-skilled individuals who could be moved to resolve bottlenecks when they are idle. This is the best method of replicating swarming when faced with a bottleneck in the model. If the teams being used are sufficiently well analysed there is also the opportunity to specify their abilities through the resource blocks which all adds to the variability that real life would have in this regard.

Chapter 7: Conclusions

7.1 General Reflection

There is no doubting that the Kanban approach is a highly effective approach for software development. Numerous advantages to the approach have been highlighted throughout this study but its adaptability serves as a notable aspect because it could feasibly be adopted in any software process. This is the strongest of its many favourable traits because Kanban could bring some bit of an improvement to all sorts of scenarios merely by pushing its ideologies like visualisation of work. The application of Kanban to software development is merely in its infancy so there is many more discoveries to be made that will help it grow into an even more valuable tool in the future.

Business process simulation modelling is an extremely powerful and valuable tool that should be adopted by any organisation as it can bring about insights that would be almost impossible to achieve without risking or disrupting business operations. Decision making with regards various aspects of the system and even answers to "what-if?" analyses all serve to benefit organisations as well as assisting managers to run their businesses well. The cost associated with acquiring commercially available simulation software is negligible in terms of the savings that can be made through optimisation and analysis of the business process.

7.2 Research Questions Addressed

7.2.1 The Kanban Approach and Simulation Modelling

The use of ExtendSim to model the Kanban approach proved successful despite the lack of real data to achieve validity for the model. The model provided provides a strong framework from which genuine software processes could be modelled by emulating the concept proposed here or by modifying as necessary. The hierarchical abilities of the simulation software allows for the creation of a Kanban board and within each hierarchical block the process can be customised to be exclusive to the process being modelled. The ability to vary the internal workings means that there is an adaptability to differing systems while the overall Kanban board can retain its structure despite different functional elements.

The ability for simulation models to so easily capture the necessities of the Kanban approach such as visualisation of workflow, limiting WIP, and measuring lead time (Kniberg

& Skarin, 2010) means that Kanban can be easily modelled through the use of software simulation. This ease means that the true difficulty is in making a model that is valid so that accurate inferences can be drawn from its results. It has being illustrated in this study how straightforward the process of creating a simulation can be. There is a necessity for a significant investment of time in order to achieve these goals but over time adapting simulation models should be a quick job as expertise and familiarity have being heightened.

7.2.1 The Kanban Approach and the Social Sciences

The broad array of literature chosen from psychological and management literature demonstrates considerable support for the use of the Kanban approach in a software development setting. The specific applications of these theories is found throughout Chapter 3 and summarised for convenience in Section 3.7. General conclusions from this is that the numerous individual, team, and organisational benefits that emerge through the use Kanban approach are supported by the literature. What is remarkable is that Kanban essentially has three constraints that it must abide by and through visualising the workflow, limiting WIP, and measuring lead time (Kniberg & Skarin, 2010) yet much of psychological literature cites the benefits of doing such things, though not these specific constraints. Management literature is predominantly finding an affinity with the influences of agile as this places a great deal of attention upon the individual and the collaborative elements. The elements that Kanban has inherited from agile are of high value in a management context. Overall, the Kanban approach seeks to be more than just a method of operational efficiency and the research observed from social science demonstrates the broad set of beliefs that it possesses for the benefit people in particular.

7.3 Recommendations for Further Research

This research has been limited by the absence of any real data from a software development process so that the simulation model provided could be validated. It would necessary for any future research in a similar vein to achieve model validation before any additional progress could be made to broaden the scope of research.

Before completing this research the complementary nature of simulation software and the Kanban approach was not truly realised. It would certainly be of future interest to examine the value of attempting to model a software process through the Kanban approach

whilst also modelling the process with simulation software. Simulation software is considerably detailed whilst the Kanban approach manages to make a complex system appear less so thanks to the Kanban board.

On the point of completing simulations, it would be valuable to gain some degree of quantifiable insight into how difficult various software industry employees find learning to simulate software processes. With such knowledge there would be a guideline for managers to weigh up training some of their employees and also which employees could be most capable of learning to do so based on their traits and skills.

References

Anderson, D. J. (2010). *Kanban: Successful evolutionary change for your technology business*. USA: Blue Hole Press.

Anderson, D. J., & Roock, A. (2011). An Agile evolution: Why Kanban is catching on in Germany and around the world. In *Viral Growth of Kanban in the Enterprise*, 6-11.

Anderson, D., Concas, G., Lunesu, M. I., & Marchesi, M. (2011) Studying Lean-Kanban Approach Using Software Process Simulation. In *Agile Processes in Software Engineering and Extreme Programming*, 12-26. Springer Berlin Heidelberg.

Benson, J. (2012). The Psychology of Kanban. [Online Video]. Available from: http://vimeo.com/30696302 (Accessed 16 April, 2014)

Banks, J., Carson, J. S., Nelson, B. L., & Nicol, D. M. (2004). *Discrete-Event System Simulation* (4th ed.). New Jersey: Prentice-Hall Inc.

Beer, M., & Nohria, N. (2000) Cracking the Code of Change. *Harvard Business Review,* 78(3), 133-141.

Cassettari, L., Gallo, M., Montella, D. R., Revetria, R., Romano, E., & Testa, A. (2012). Analysis of demand variability in a kanban cell by virtual Kanban strategy. In *Proceedings of the 6th WSEAS international conference on Computer Engineering and Applications, and Proceedings of the 2012 American conference on Applied Mathematics* (pp. 366-370). World Scientific and Engineering Academy and Society (WSEAS)

Cocco, L., Mannaro, K., Concas, G., & Marchesi, M. (2011). Simulating Kanban and Scrum vs. Waterfall with System Dynamics. In *Agile Processes in Software Engineering and Extreme Programming* (117-131). Springer Berlin Heidelberg.

Cockburn, A., & Highsmith, J. (2001). Agile software development: The people factor. *Computer, 34*(11), 131-133.

Colom, R., Martínez-Molina, A., Shih, P. C., & Santacreu, J. (2010). Intelligence, working memory, and multitasking performance. *Intelligence, 38*(6), 543-551.

Corona, E., & Pani, F. E. (2013). A Review of Lean-Kanban Approaches in the Software Development. *WSEAS Transactions on Information Science & Applications*, *10*(1).

Csikszentmihalyi, M. (2004). *Good business: Leadership, flow, and the making of meaning*. USA: Penguin.

Deming, W. E. (2000). *The new economics: for industry, government, education*. Cambridge: MIT press.

Dixon, J., & Jones, T. (2011). Hype cycle for business process management, 2011. *Gartner Research*, 1-89.

Fowler, M., & Highsmith, J. (2001). The agile manifesto. *Software Development*,*9*(8), 28-35.

Goldratt, E. M., & Cox, J. (1993). *The goal: a process of ongoing improvement* (Vol. 2). Aldershot: Gower.

Hoch, J. E., Pearce, C. L., & Welzel, L. (2010). Is the Most Effective Team Leadership Shared?. *Journal of Personnel Psychology*, *9*(3), 105-116.

Ikonen, M., Pirinen, E., Fagerholm, F., Kettunen, P., & Abrahamsson, P. (2011). On the impact of kanban on software project work: An empirical case study investigation. In *Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on*, 305-314.

Ingason, H. T., Gestsson, E., & Jonasson, H. I. (2013). The Project Kanban Wall: Combining Kanban and Scrum for Coordinating Software Projects. *PM World Journal, 2,* 1-23.

Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research*, *203*(1), 1-13.

Kauffeld, S. (2006). Self-directed work groups and team competence. *Journal of Occupational and Organizational Psychology*, *79*(1), 1-21.

Kellner, M. I., Madachy, R. J., & Raffo, D. M. (1999). Software process simulation modeling: Why? What? How?. *Journal of Systems and Software*, *46*(2), 91-105.

Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum-making the most of both*. USA: C4Media Inc.

Kniberg, H. (2011). *Lean from the trenches: Managing large-scale projects with Kanban*. USA: Pragmatic Bookshelf.

Konig, C. J., Buhner, M., & Murling, G. (2005). Working memory, fluid intelligence, and attention are predictors of multitasking performance, but polychronicity and extraversion are not. *Human Performance*, *18*(3), 243-266.

Laanti, M. (2013). Agile and Wellbeing - Stress, Empowerment, and Performance in Scrum and Kanban Teams. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, 4761-4770.

Laguna, M., & Marklund, J. (2005). *Business Process Modeling, Simulation and Design*. New Jersey: Pearson Education.

Langfred, C. W. (2005). Autonomy and performance in teams: The multilevel moderating effect of task interdependence. *Journal of management*, *31*(4), 513-529.

Lunesu, M. I. (2013). Process software simulation model of Lean-Kanban approach (Unpublished doctoral dissertation). University of Cagliari, Cagliari.

Mahnic, V. (2014) Improving Software Development through Combination of Scrum and Kanban. In *Recent Advances in Computer Engineering, Communications and Information Technology,* 281-288.

Masicampo, E. J., & Baumeister, R. F. (2011). Consider it done! Plan making can eliminate the cognitive effects of unfulfilled goals. *Journal of personality and social psychology*, *101*(4), 667.

Middleton, P., & Joyce, D. (2012). Lean software management: BBC Worldwide case study. *Engineering Management, IEEE Transactions on*, *59*(1), 20-32.

Näslund, D. (2008). Lean, six sigma and lean sigma: Fads or real process improvement methods?.,*Business Process Management Journal*, *14*(3), 269 – 287.

Norrmalm, T. (2011). Achieving Lean Software Development: Implementation of Agile and Lean Practices in a Manufacturing-Oriented Organization (Unpublished doctoral dissertation). Uppsala University, Uppsala.

Poppendieck, M., & Poppendieck, T. (2003). *Lean software development: an agile toolkit*. Boston: Addison-Wesley Professional.

Rautiainen, K. (2010). Chapter 12: Kanban for Software Development. In V. Heikkilä, K. Rautiainen & J. Vähäniitty (Eds.) *Towards Agile Product and Portfolio Management*, 184-192.

Reese, L., & Iberle, K. (2011) Kanban – What is it and why should I care?. *Pacific Northwest Software Quality Conference 2011*, 1-11.

Rus, I., Neu, H., & Münch, J. (2003). A systematic methodology for developing discrete event simulation models of software development processes. In *Proceedings of the 4th International Workshop on Software Process Simulation and Modelling*..

Sjøberg, D. I., Johnsen, A., & Solberg, J. (2012). Quantifying the effect of using kanban versus scrum: A case study. *Software, IEEE*, *29*(5), 47-53.

Sugimori, Y., Kusunoki, K., Cho, F., & Uchikawa, S. (1977). Toyota production system and kanban system materialization of just-in-time and respect-for-human system. *The International Journal of Production Research*, *15*(6), 553-564.

Tannenbaum, R., & Schmidt, W. H. (1973). How to choose a leadership pattern. *Harvard Business Review*.

Terlecka, K. (2012). Combining Kanban and Scrum - Lessons from a Team of Sysadmins. In *Agile Conference (AGILE), 2012*, 99-102.

Turner, R., Madachy, R., Ingold, D., & Lane, J. A. (2012). Modeling kanban processes in systems engineering. In *Software and System Process (ICSSP), 2012 International Conference on*, 23-27.

Van der Aalst, W. M., Nakatumba, J., Rozinat, A., & Russell, N. (2008). Business Process Simulation: How to get it right. *BPM Center Report*, 1-25.

Vermeulen, F., Puranam, P., & Gulati, R. (2010). Change for change's sake. *Harvard Business Review*, *88*(6), 70-76.

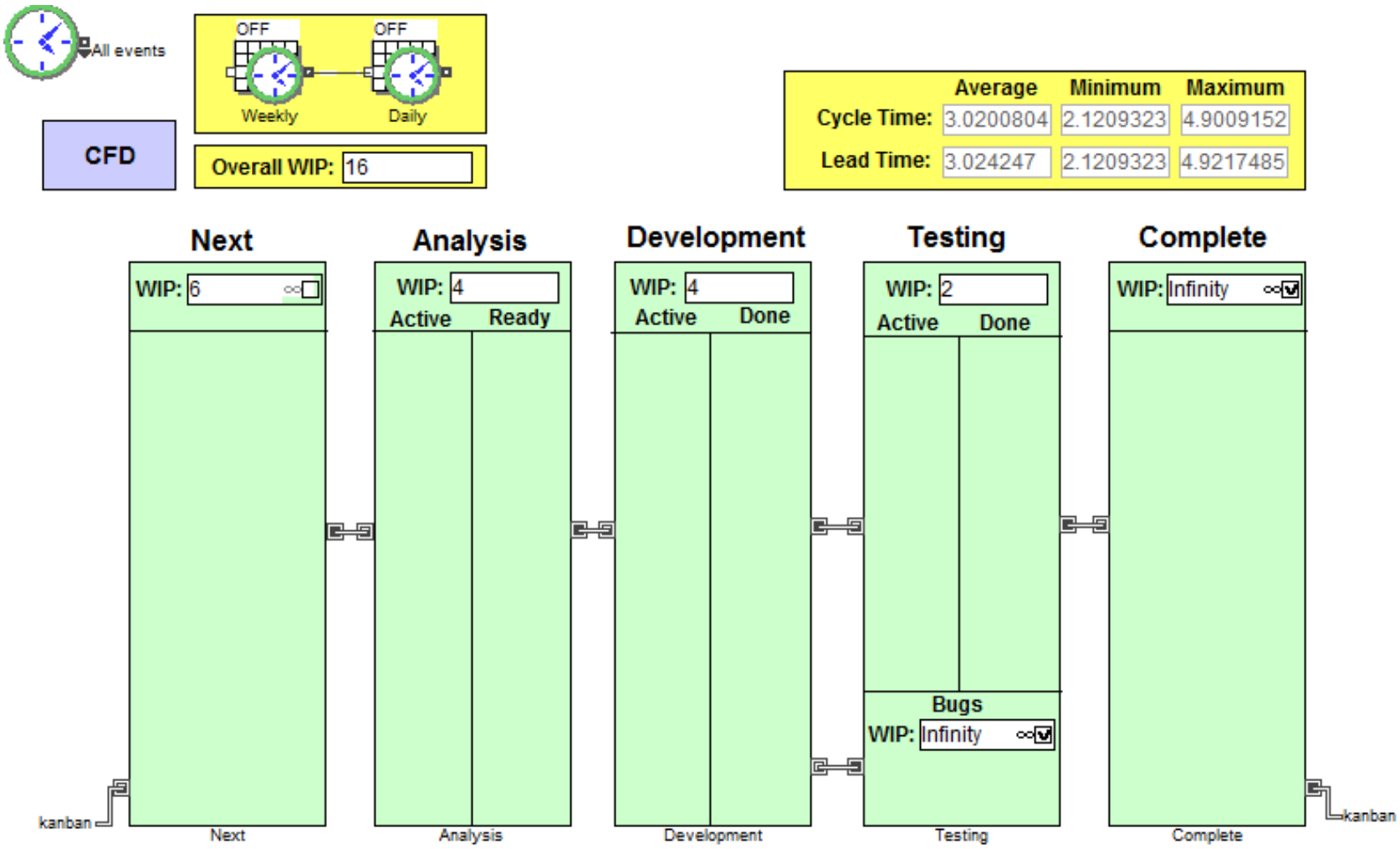
Weinberg, G. M. (1992). *Quality software management (Vol. 1): Systems thinking*. New York: Dorset House Publishing Co. Inc.

West, D., Grant, T., Gerush, M., & D'Silva, D. (2010). Agile development: Mainstream adoption has changed agility. *Forrester Research*, *2*, 41.

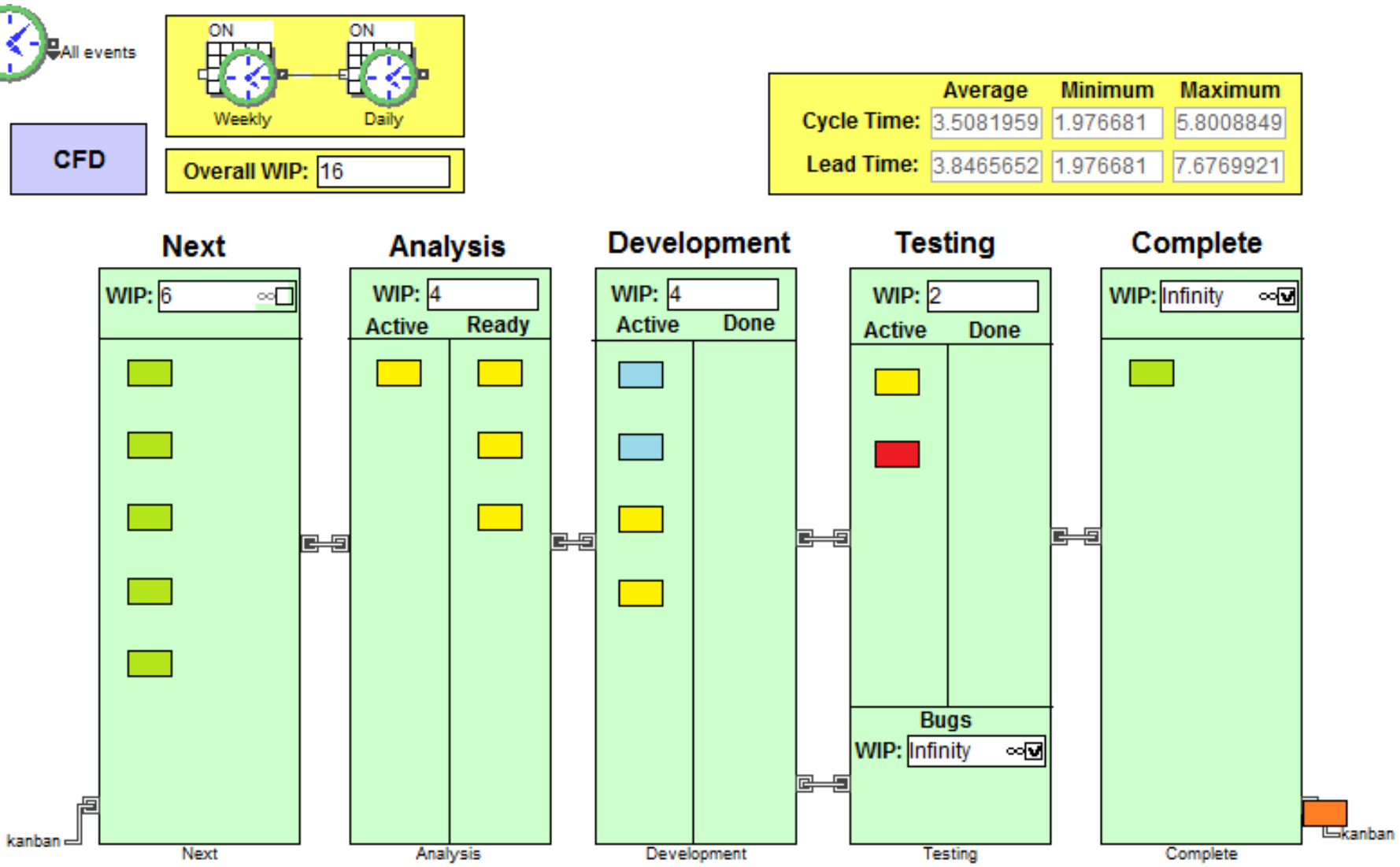
Womack, J. P., Jones, D. T., & Roos, D. (1990). *The machine that changed the world*. USA: Harper Collins.

Wong, Z. (2007). *Human factors in project management: concepts, tools, and techniques for in*f unfulfilled goals. *Journal of personality and social psychology*, *101*(4), 667.

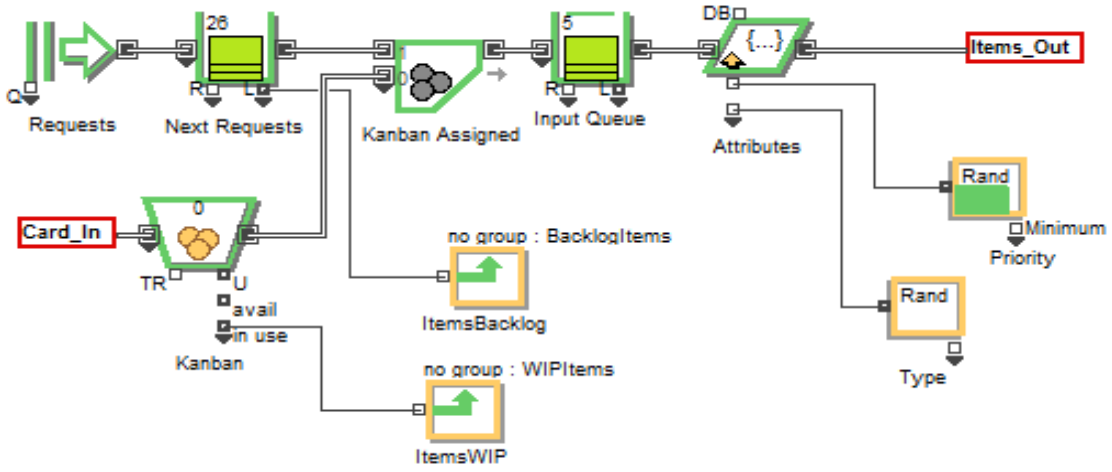
Appendices



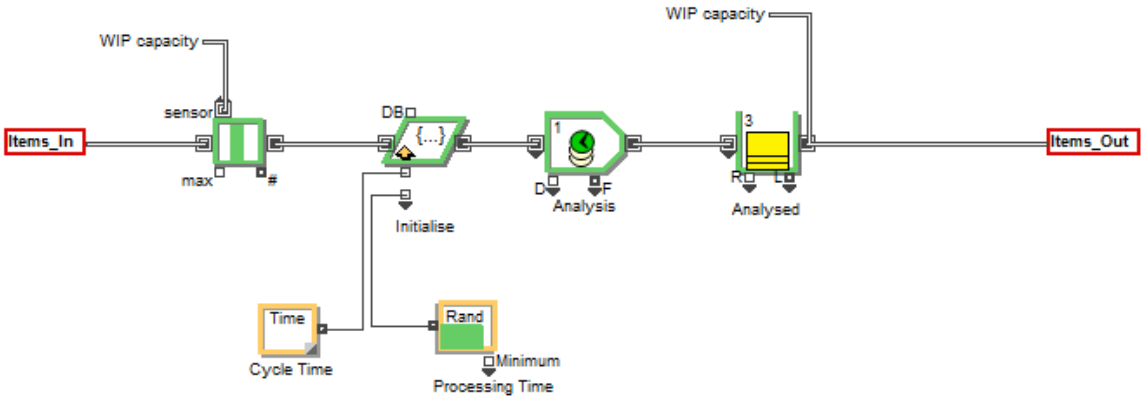
Appendix A: Kanban model developed in ExtendSim to imitate a Kanban board



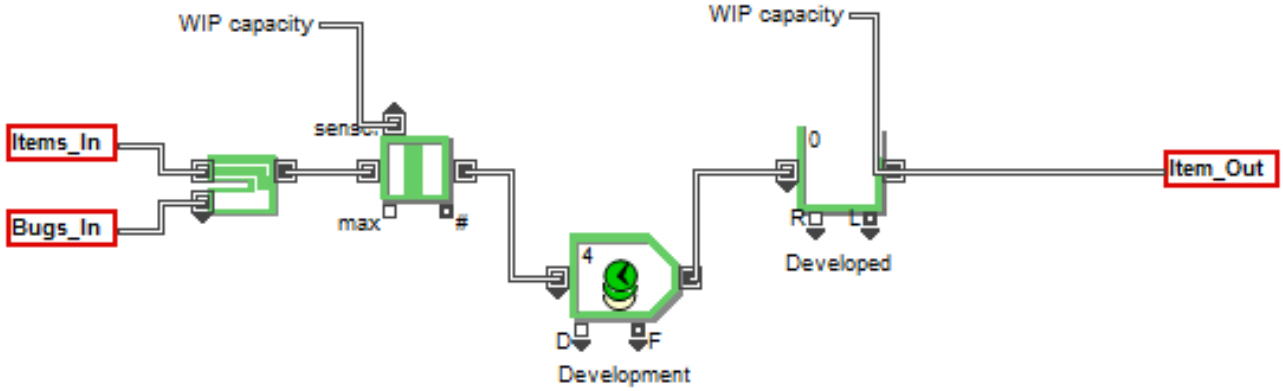
Appendix B: An active image of the Kanban model which illustrates the task cards moving along the Kanban board



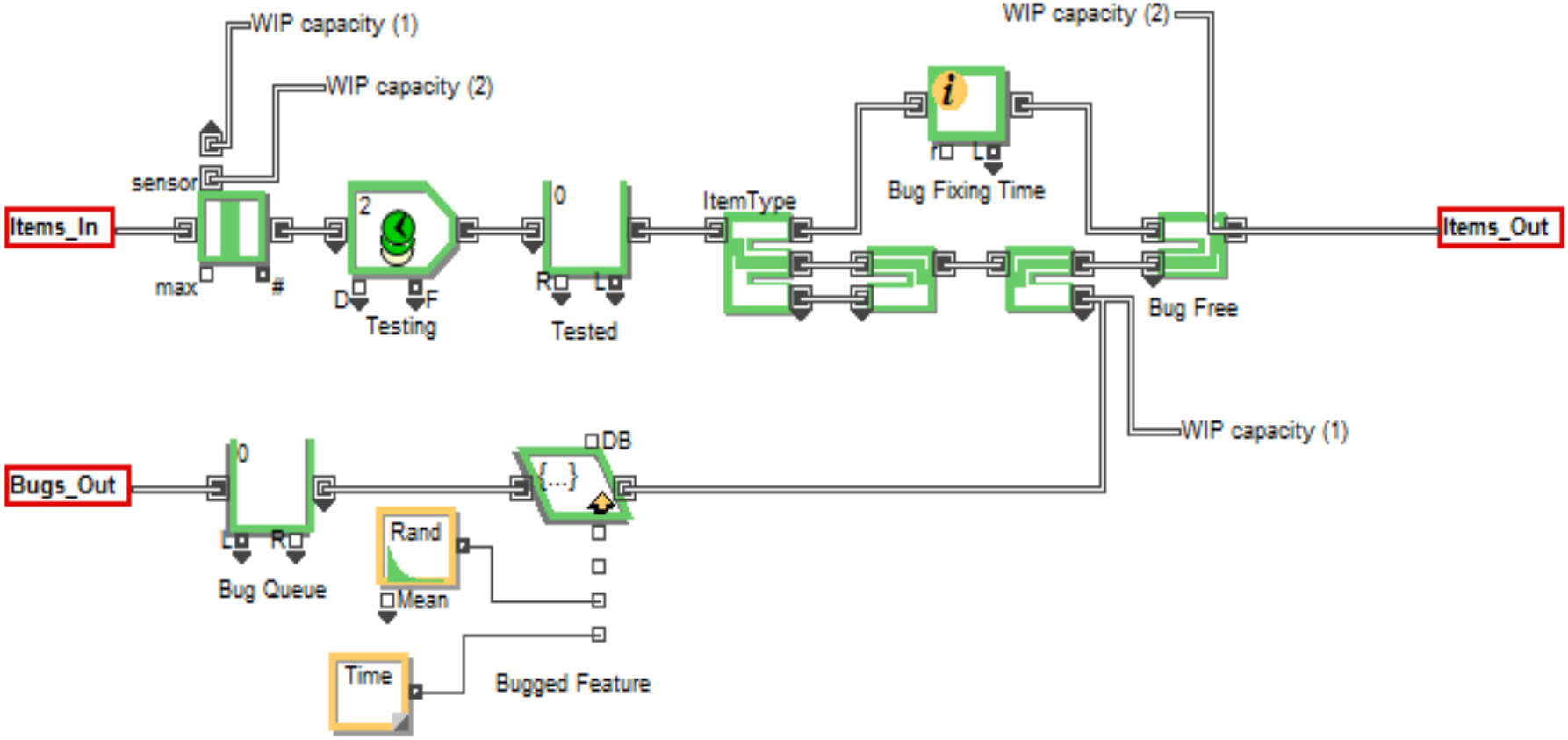
Appendix C: The blocks running inside the hierarchy of the Next stage



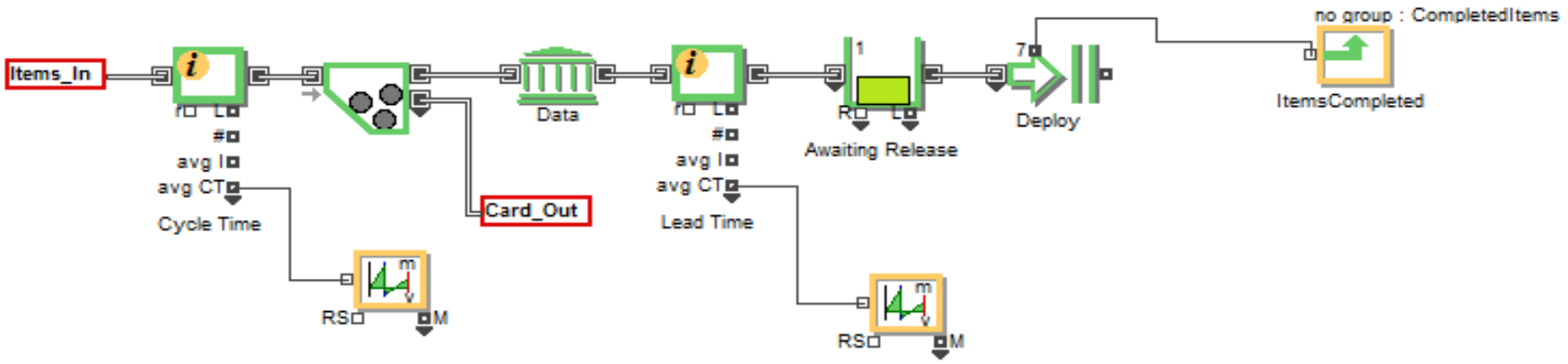
Appendix D: The blocks running inside the hierarchy of the Analysis stage



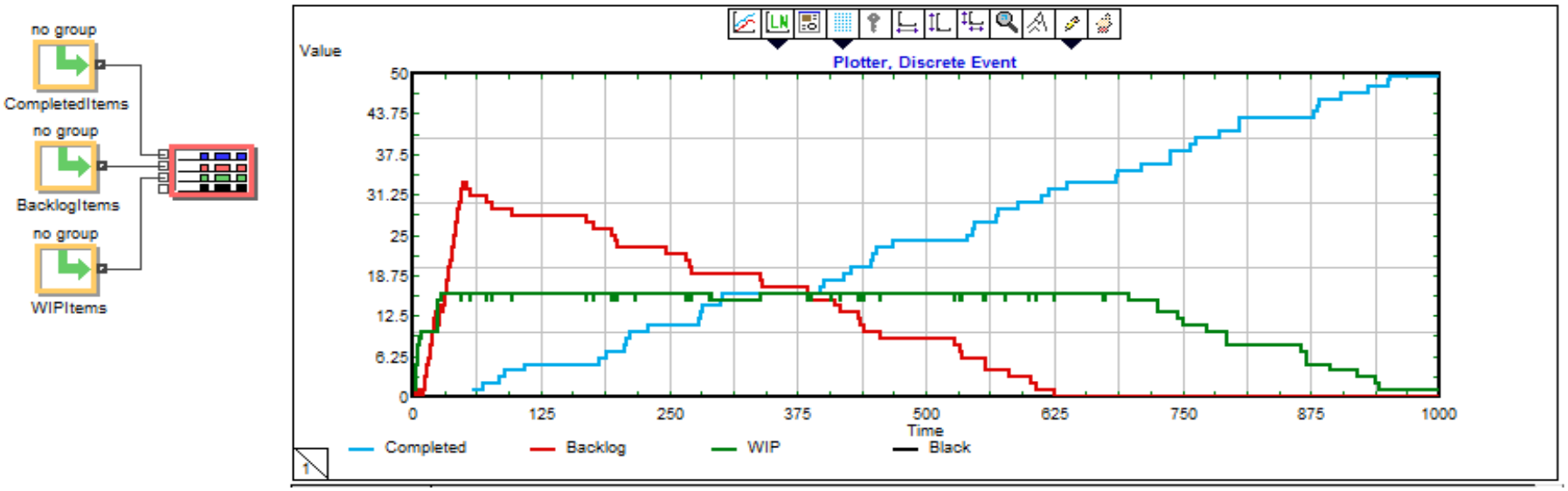
Appendix E: The blocks running inside the hierarchy of the Development stage



Appendix F: The blocks running inside the hierarchy of the Testing stage of the Kanban board



Appendix G: The blocks running inside the hierarchy of the Completed stage of the Kanban board



Appendix H: The blocks running inside the hierarchy of the CFD block alongside the Kanban board