**CS4348 Software Quality Management, Semester B, 2021/2022**

**Assignment 2 Submission by Group 2**.

**Study report of agile software process model (named: Kanban ).**

**Contribution of Each Group Member:**

*\* The actual amount of time spent is only used for statistics and will NOT affect the marks awarded. EXCLUDE the time you spent in background reading and/or revision study of the course materials.*

| **Student ID** | **Full (short) name of the author** | **Parts/Sections**  **written/presented by the team member** | **Time spent (hours)\*** |
| --- | --- | --- | --- |
| 54051066 | Au Ka Fu | * Section 0. Teamwork process * Section 1.2 SQA, SQM and Software Life-Cycle * Section 4. Presentation scripts (Part 1) * Part 2 of powerpoint slides (number 1 to 23) | 12 |
| 55241185 | Leung Man Yuen | * Section 0. Teamwork process * Section 1.4. Practice Conforming Software Quality * Section 2.2. Benefits and Difficulties for Kanban * Section 2.3. Comparing Kanban with Other Models * Section 3.1 SQA/SQM Standards from IEEE 12207-2017 with Kanban | 18 |
| 56267334 | Lam Ka Yik | * Section 0. Teamwork process * Section 1.3. Summarizing the course materials relevant to Kanban (title of subsection) * Section 2.1. The essentials of the selected process model (title of subsection) * Section 3.2 SQM activities and practices with Kanban * Section 3.3 Effectiveness in SQM with Kanban * Section 3.4 How Kanban adheres to the general Agile Manifesto | 50 |
| 56262676 | Chung Ho Kin | * Section 0. Teamwork process * Section 4. Presentation scripts (Part 2) * Part 2 of powerpoint slides (number 24 to 43) | 15 |
| 55882855 | Chan Shing Hei | * Section 0. Teamwork process * Section 1.5 * Section 5. Software tool * References | 9 |

**Table of Content**

| **Section / Subsection** | **Author name (** Signature **)** | **Reviewer name (** Signature **)** |
| --- | --- | --- |
| * Section 0. Teamwork process | All | Lam Ka Yik  (Ken Lam) |
| * Section 1. Background |  |  |
| – Section 1.1. Overview of Kanban | Chung Ho Kin  (Ken Chung) | Leung Man Yuen  (Leung) |
| – Section 1.2. SQA, SQM and Software Life-Cycle | Au Ka Fu  (Kidd Au) | Chung Ho Kin  (Ken Chung) |
| – Section 1.3. Summarizing the course materials relevant to Kanban | Lam Ka Yik  (Ken Lam) | Au Ka Fu  (Kidd Au) |
| – Section 1.4. Practices Conforming Software Quality | Leung Man Yuen  (Leung) | Chan Shing Hei  (Rena Chan) |
| – Section 1.5. More concepts on the Software Quality | Chan Shing Hei  (Rena Chan) | Lam Ka Yik  (Ken Lam) |
| * Section 2. Description of the process model |  |  |
| – Section 2.1. Description of the process model | Lam Ka Yik  (Ken Lam) | Leung Man Yuen  (Leung) |
| – Section 2.2. Benefits and Difficulties for Implementing Kanban | Leung Man Yuen  (Leung) | Chan Shing Hei  (Rena Chan) |
| – Section 2.3. Comparing Kanban with Other Models | Leung Man yuen  (Leung) | Lam Ka Yik  (Ken Lam) |
| * Section 3. SQA/SQM activities/tasks |  |  |
| – Section 3.1. SQA/SQM Standards from IEEE 12207-2017 with Kanban | Leung Man Yuen  (Leung) | Lam Ka Yik  (Ken Lam) |
| – Section 3.2. SQM activities and practices with Kanban | Lam Ka Yik  (Ken Lam) | Chan Shing Hei  (Rena Chan) |
| – Section 3.3. Effectiveness in SQM with Kanban | Lam Ka Yik  (Ken Lam) | Leung Man Yuen (Leung) |
| – Section 3.4. How Kanban adheres to the general Agile Manifesto | Lam Ka Yik  (Ken Lam) | Chan Shing Hei  (Rena Chan) |
| * Section 4. Presentation scripts (and slides in a separate file) |  |  |
| – Part 1. Slide numbers from 1 to 23 | Au Ka Fu  (Kidd Au) | Chung Ho Kin (Ken Chung) |
| – Part 2. Slide numbers from 24 to 43 | Chung Ho Kin  (Ken Chung) | Au Ka Fu  (Kidd Au) |
| * Section 5. Software tool | Chan Shing Hei  (Rena Chan) | Chung Ho Kin (Ken Chung) |
| * References | All | Au Ka Fu  (Kidd Au) |

**Section 0. Teamwork process**

**Action Plan**

| **Task ID** | **Dependency** | **Task** | **Start** | **End** | **Duration(Days)** |
| --- | --- | --- | --- | --- | --- |
| 1 | Nil | Meeting for selecting software process models | 13/3/2022 | 13/3/2022 | 1 |
| 2 | 1 | Voting on agile software process models topic | 14/3/2022 | 14/3/2022 | 1 |
| 3 | 2 | Sending email | 15/3/2022 | 16/3/2022 | 2 |
| 4 | 3 | Meeting for Distribution of Section 2-5 | 22/3/2022 | 22/3/2022 | 1 |
| 5 | Nil | Meeting for Complete Section 0 & Discussion and Distribution of Section 1 | 26/3/2022 | 26/3/2022 | 1 |
| 6 | 5 | Complete and Review Section 1 | 26/3/2022 | 30/3/2022 | 5 |
| 7 | 6 | Revise Section 1 | 30/3/2022 | 31/3/2022 | 2 |
| 8 | 4,5 | Complete and Review Section 2 | 26/3/2022 | 2/4/2022 | 8 |
| 9 | 8 | Revise Section 2 | 2/4/2022 | 3/4/2022 | 1 |
| 10 | 9 | Complete and Review Section 3 | 4/4/2022 | 6/4/2022 | 3 |
| 11 | 10 | Revise Section 3 | 6/4/2022 | 7/4/2022 | 2 |
| 12 | 4,7 | Start Section 4 Presentation slide & scripts | 31/3/2022 | 3/4/2022 | 4 |
| 13 | 12 | Review and revise Slide & Script of Section 1 | 3/4/2022 | 4/4/2022 | 2 |
| 14 | 9,13 | Review and revise Slide & Script of Section 2 | 4/4/2022 | 7/4/2022 | 4 |
| 15 | 11,14 | Review and revise Slide & Script of Section 3 | 7/4/2022 | 10/4/2022 | 4 |
| 16 | 4,5 | Review and revise Section 5 | 24/3/2022 | 9/4/2022 | 15 |
| 17 | 16 | Revise Section 5 | 9/4/2022 | 10/4/2022 | 1 |
| 18 | 17 | Review and revise Slide & Script of Section 5 | 10/4/2022 | 12/4/2022 | 3 |
| 19 | 18 | Review reference and Submit Assignment 2 | 12/4/2022 | 13/4/2022 | 2 |

**Task description**:

1. Meeting for selecting agile software process models
   1. Deadline: 13/3/2022
   2. Venue: Online meeting - Discord
   3. Content:
      1. Discussing to suggest agile software process models
2. Voting on agile software process models topic
   1. Deadline: 14/3/2022
   2. Venue: Online meeting - Discord
   3. Content: Make the decision which is top 3 agile software process models
3. Sending email
   1. Deadline: 16/3/2022
   2. Venue: Outlook
   3. Content:
      1. Sending email for selecting the software process model to professor
4. Meeting for Distribution of Section 2-5:
   1. Deadline: 22/3/2022
   2. Venue: Online meeting - Discord
   3. Content: Distribution of works-
      1. Section 2. Description of the process model (Ken Lam + Leung)
      2. Section 3. SQA/SQM activities/tasks (Ken Lam + Leung)
      3. Section 4. Presentation scripts (Kidd + Ken Chung)
      4. Section 5. Software tool (Hei)
      5. Distribution of Section 2-5 works with teammate
5. Meeting for Complete Section 0 & Discussion and Distribution of Section 1:
   1. Deadline: 26/3/2022
   2. Venue: Discord
   3. Content:
      1. Complete Section 0
      2. Discuss the potential parts in Section 1
      3. Distribute the agreed parts in Section 1
6. Review Section 1 and add comment
   1. Deadline: 30/3/2022
   2. Venue: Google document
   3. Content:
      1. Leung Man Yuen reviews and comments on section 1.1
      2. Chung Ho Kin reviews and comments on section 1.2
      3. Au Ka Fu reviews and comments on section 1.3.
      4. Chan Shing Hei reviews and comments on section 1.4.
      5. Lam Ka Yik reviews and comments on section 1.5.
7. Review Section 2 and add comment
   1. Deadline: 2/4/2022
   2. Software: Google document
   3. Content:
      1. Leung Man Yuen reviews and comments on section 2.1.
      2. Chan Shing Hei reviews and comments on section 2.2.
      3. Lam Ka Yik reviews and comments on section 2.3.
8. Review Section 3 and add comment
   1. Deadline: 6/4/2022
   2. Software: Google document
   3. Content:
      1. Lam Ka Yik reviews and comments on section 3.1.
      2. Chan Shing Hei reviews and comments on section 3.2.
      3. Leung Man Yuen reviews and comments on section 3.3.
      4. Chan Shing Hei reviews and comments on section 3.4.
9. Review Section 4 and Slide and then add comment
   1. Deadline: 10/4/2022
   2. Software:
      1. Google powerpoint
      2. Google document
   3. Content:
      1. Chung Ho Kin reviews and comments on section 4 Part 1.
      2. Au Ka Fu reviews and comments on section 4 Part 2.
10. Review Section 5 and add comment
    1. Deadline: 9/4/2022
    2. Software: Google document
    3. Content:
       1. Lam Ka Yik reviews and comments on section 5
11. Review References
    1. Deadline: 12/4/2022
    2. Venue: Google document
    3. Content:
       1. Au Ka Fu reviews references part

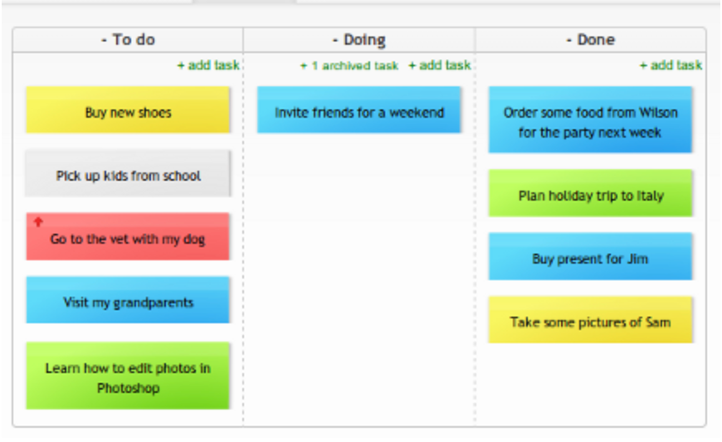
**Section 1. Background**

**Section 1.1 Overview of Kanban**

Taiich Ohno introduced Kanban in the early 1940s[1]. He was an Industrial Engineer and Businessman in Toyota Motor Corporation which is a Japanese multinational automotive manufacturer headquartered in Toyota City, Aichi, Japan. The Kanban was developed to increase the productivity and efficiency of the production control system while reducing the cost-intensive inventory of materials and final products. The Kanban is helpful to manufacture inductors because it can mitigate the impact of supply disruption and overstocking of goods at various stages of the manufacturing process. However, it requires attention and continuous monitoring of the process to avoid bottlenecks. Otherwise, the production process will slow down and be delayed for the final products.

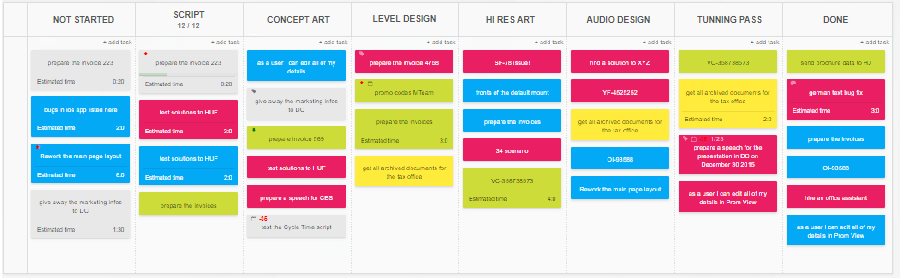
Kanban was introduced to software development by David J. Anderson in 2004. He is a thought leader in effective software development and published a book called “Kanban : Successful Evolutionary Change for Your Technology Business”[2] . The simplest Kanban software development needs a board consisting of 3 columns: Backlog, In Progress and Done. The board helps the software team to visualize the overview of their process and helps to identify and resolve any workflow problems. Also, it's easy to follow up and collaborate with different project team members.

Kanban has high flexibility and can be changed into different formats depending on your needs. The following are some examples:



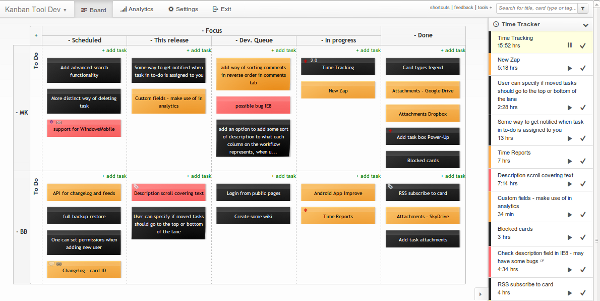
**Figures 1.1 Basic Kanban [3]**

This is a simple structure of Kanban; it contains the "To do", "Doing/In Progress", and "Done" columns. It helps you to visualize the whole team's work, to standardize the workflow, and to identify any obstacles or dependencies to solve problems quickly.



**Figures 1.2 Heijunka board[4]**

Heijunka is a Japanese word which means production leveling. The principle of the Heijunka is to level production, create stability and predictability. The above example is used by game developers, the tasks are parallel execution which means every developer should have a task and the board helps to control the rate of workload.



**Figures 1.3Time Tracking in software development[5]**

This example is about time tracking tools that helping software teams improve their development processes, deliver their products, and meet customers' needs. The board column has To-do, Schedule, This release, Dev Queue, In progress and Done. There are three types of cards to represent different tasks: black for user requests, red for bugs, and orange for valuable ideas for new additions or new features. Each row of the board represents a developer. The board's purpose is to provide statistical data and summaries of each worker's progress and project workflow.

**Section 1.2 SQA, SQM and Software Life-Cycle**

What is SQA?

Software Quality Assurance (SQA) is a process that monitors all methods, other software processes, activities in a project to ensures the quality of software that conforms to the current standards, including CMMI, ISO 9000, etc.

SQA is an ongoing process in the Software Development Life Cycle (SDLC) that periodically checks developed software to guarantee that it achieves the required quality measures. The main objective of the SQA process is to reduce the risk of the development process and enhance the software quality with a certain time and budget.

What is SQM?

Software Quality Management (SQM) is a management aspect that control and guarantee the quality of a software can achieve the standards and also meets the project requirements, which is a process that throughout the whole development life cycle.

What is Software life-cycle?

Software life-cycle, also known as Software Development Life Cycle (SDLC), which is a terminology used by software engineers and developers to describe the process of planing, designing and developing a software. The SDLC concept covers the process of the planning, creating, testing and final deploying a software. The SDLC can be applied to both hardware and software, which commonly includes 6 stages: Requirements Gathering and Analysis, Design, Implementation, Testing, Deployment, and Maintenance. Some developers will combine or separate some stages based on necessity.

The biggest advantage using Kanban in SQM process is that it allows developers to visualize the work flow of the software development. Since the best process for different systems is not the same, the lifecycle of a Kanban can also vary from system to system. Therefore, the most appropriate way to classify the lifecycle of Kanban is according to the feedback loops involved.

Besides the visualization of works, there are many implementations of Kanban that are able to help in SQM and SQA processes. For example, limiting work in progress, making policies explicit, conducting feedback loops, etc. Kanban is applicable when there are unexpected works in the development process, or work parallelly without waiting for the tasks with dependency finished. It fulfills the standards that shorten the time spent in the development process, at the same time not affect the software quality.

**Section 1.3. Summarizing the course materials relevant to Kanban**

Kanban is a lightweight methodology which provides a life cycle to organize and handle the tasks and processes. It helps us implement the software product with a systematic approach. It has 2 parts which are lead time and cycle time. Lead time means the task is placed into the product backlog and before delivering the product to the customer. The cycle time means the task is started and put into the Kanban board until the end of the tasks. Also, following the workflow of the Kanban, we can manage the project process easily. In the planning phase, the Kanban can identify the priority-order product features and have a schedule which can apply maximum utility manpower. In the development phase, Kanban distributes the project to a few subprojects as a Kanban card and then allot to different teammates to finish it. Kanban is responding to change rather than following the schedule. It lets customers change the requirement and improve the process with regular review of the process of the Kanban.

Applying Kanban means that the manager can review the state of the development easily. Kanban offers teams a visualized process flow with Kanban boards and cards. All cards in the Kanban board represent processes and have different status to remind the team of what the teammate needs to handle the task first. Providing a good management review is one of the benefits when we use Kanban.

Kanban provides a team with a user-friendly environment to do the peer review such as Team Review, pair programming, Passaround and so on. Kanban lists all the tasks in the Kanban board. Therefore, the other teammates can easily know what your teammates are doing. Also, the card details can mention who needs to review the tasks. After the teammates reviews the tasks and ensures the quality of the product, the card can be moved to the next status. Applying Kanban is useful for a project team to do the peer review.

Kanban can handle a part of the project constraints well such as people, quality and schedule. Having work-in-progress limits, maximum utility manpower will be achieved. A person has lots of jobs and then the project manager will not allot the task to him or her, while the person finishes their job and has a quota of the work-in-progress limits and then the manager can do manpower deployment to enhance the effectiveness. Applying the Kanban should not pass the low quality product to the team. Therefore, they are able to defer commitment if they do not have enough time to finish it. Moreover, the Kanban task can be reviewed before submitting the product to the team. It can ensure the quality of the product. With a visualized workflow, the project schedule will be clear enough and it allows deferring commitment. Therefore, it has a flexible schedule compared to other process models.

With Kanban, the clear and detailed workflow is shown to the team. Software configuration item is easy for the team to handle after we list what we need to check and update in the Kanban card. Checking the version of the document, code of the product, testing data and the development tools should be checked before handing out the product. If the software configuration is incorrect, it may make the whole project inconsistent or have mistakes in the product. For example, using the old source code to test the latest test cases does not work. It will cause trouble to other teammates when they do the next step of the product. Therefore, Kanban can offer an effective way for a team to manage the software configuration.

Kanban can offer a convenient way for measurement to a project team to enhance the product such as product metrics, process metrics and project metrics. During the development of the product, when a task is finished, the peer review action will be taken. It can measure the quality of the project. If the quality is low, the Kanban card will not move to the next status. Therefore, Kanban can help the team to improve products. Also, it can help us to reallocate resources to reduce the implementation time of the project.

**Section 1.4 Practices Conforming Software Quality**

Project management is the methodology of applying principles and practices during different phases in a project. These phases include planning, executing, monitoring and controlling and closing. A software project management plan should specify the goals and relevant technical and managerial processes to meet the requirements. Technical management processes and project management processes play a significant role in developing the final products. Kanban can help to evaluate processes by adopting a plan-do-check-act approach. For every process, a team may list the task on the Kanban and trace the result to see if it has to be dropped or adopted in future plans. Each project can be decomposed to activities and activities can be decomposed to tasks. Moreover, Kanban is suitable for every task that can be tracked, so the approach can apply to the whole project cycle.

For quality Assurance (QA), possible activities include evaluation for the product and the development process, record for the evaluation result and writing proposed solutions for the problems that occur. IEEE 12207-2017 specify standards for software life cycle processes. The document outlines the purposes, outcomes and activities to achieve the outcomes for each task. Since these requirements are specified clearly, a development team can use Kanban to track and tell if the software conforms to the IEEE software quality standard.

Assessment and control, quality assurance, risk management and configuration management are important aspects for the technical management process. Quality management process and quality assurance process will be discussed in section 3. Risk management and configuration management will be discussed in the rest of this section.

Risk management is a practice to handle foreseeable events and unforeseeable events. Uncertainties may present as an opportunity or a loss. Changes of project constraints affect the project risk. For example, changing scope and requirements affect workload for the project and changing in budget, input and human resources affect the time and process quality of the final product. Risks can be handled by methods like prevention, mitigation or creating a contingency plan. Mitigation can be achieved through transfer of the risk to other entities. Kanban as a great process management tool can adapt to the change of process and monitor the whole situation. A proper implementation should include all necessary and important tasks to do, allowing the development team to weigh the available resources to examine and prioritize each task.

Configuration management handles different versions of configuration. A configuration composed of many configuration items. Items include software and documentation, data and developing tools for the software. If a configuration is formally reviewed and approved, it becomes a baseline. For a complex project, configuration management can be a challenge. If the version of a configuration item is mistakenly recorded, some functions or test cases may not be available. Kankan is useful when creating, updating or reviewing configuration items using a configuration management tool, such as git. Kanban provides a visualization tool to keep track of the review progress and mark if double check is done to affirm information is all correct and prevent all the reworks.

**Section 1.5 More concepts on the Software Quality**

I would like to enhance on the topic of Quality Assurance by introducing some concepts and action to the table:

CI/CD (Continuous Integration / Continuous Delivery) is a concept that describe: a set of automated task that ensure the code changes from different developer can be integrated together without any issue (CI), and a set of automated task that package said changes to customer or other user such as end-to-end tester (CD). As different developer may slowly diverge their interfaces between iterations, having a set of automated task will have benefits like: Improve software quality by introducing a set of project-wide standardized testing procedure and avoiding human error (by putting human out of the testing and deploying procedure), reduce the potential cost of fixing the software late in the development cycle (by discovering the issue earlier). CI/CD pipeline usually consists of 3 stages: build, test and deploy, in which developers can define what kind of commands / services will be used in these stages: building the application in the build stage; running test cases in the test stage; deploying the said application to servers in the deploy stage. These pipelines can be (selectively) triggered on different circumstances: only test stage when developer commits (pushing) their code to repository; build, test, and deploy when merging a branch. However, this requires the project team to set up this pipeline in the beginning of the development cycle for it to be effective, and maintain one more piece of software component afterwards (maybe adding new test cases when a new edge case is found).

**Section 2. Description of the process model**

**Section 2.1 The essentials of the selected process model**

The Kanban process model is a framework for implementing agile and DevOps software development[[1]](#footnote-0). It can visualize software project workflow, limit work in progress , continuously improve the project and focus on process flow.

To visualize the workflow, each task will be presented as a card which obtains all the necessary information. The project manager or the teammates can easily observe what work is doing or what they need to do next. It can clarify the workflow with the clear information Kanban offers.

Furthermore, the amount of work each teammate handles is shown in Kanban in order to enhance the product quality and reduce the time. Kanban boards will show the amount of tasks team members taken concurrently. Therefore, it can avoid the situation where the person handles lots of tasks and then outputs low quality products in order to meet deadlines.

Additionally, a visual workflow can make a team focus on what they need to do. Listing all the tasks presented as Kanban cards in the project can reduce the meeting time to find and discuss what is the next task they need to do. Therefore, the team focuses on process flow without interference. With reviewing the Kanban, the team can observe the real time situation of the whole project in order to prevent problems and then improve the product. It is easy for a team to discover the problem because all the tasks are measured and displayed in the Kanban. Therefore, the project manager can quickly make the decision to prevent the problem the team may make.

Kanban has a few main principles which are from change management principles and service delivery principles.

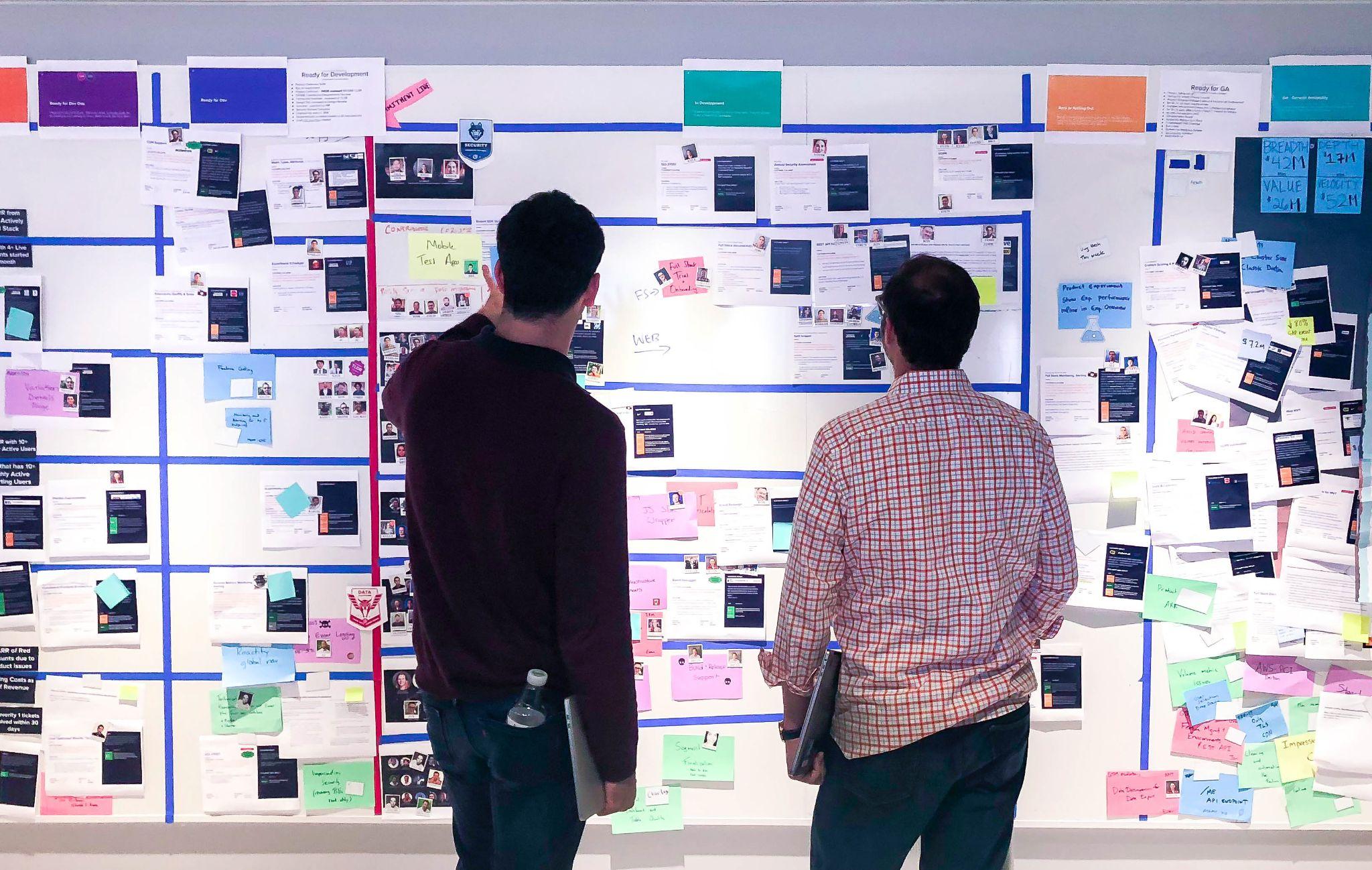
To begin with, change management principles can offer a team a new working style which is spontaneous work. Viewing the Kanban can obtain what you need to do now without any orders. It can help teammates to master skills such as how to be a good team player and a good leader with making the decisions and solving problems. It is helpful for the next person to take up the task.

Moreover, service delivery principles focus on the needs of the client and the quality of the product. Clearly understanding the requirements of the software product is important. Additionally, encouraging the staff to self-organize the work reduces the management cost. Setting rules reduce the risk of errors in order to enhance the outcome of the product. For example, a team leader needs to explain how to achieve the task after finishing the task. With explanation, the next person can understand the task and they may discover problems.

Kanban is composed of Kanban boards and Kanban cards. The Kanban board , which can be physical or digital, has 5 elements which are visual signals, columns, work-in-progress limits, a commitment point and a delivery point.

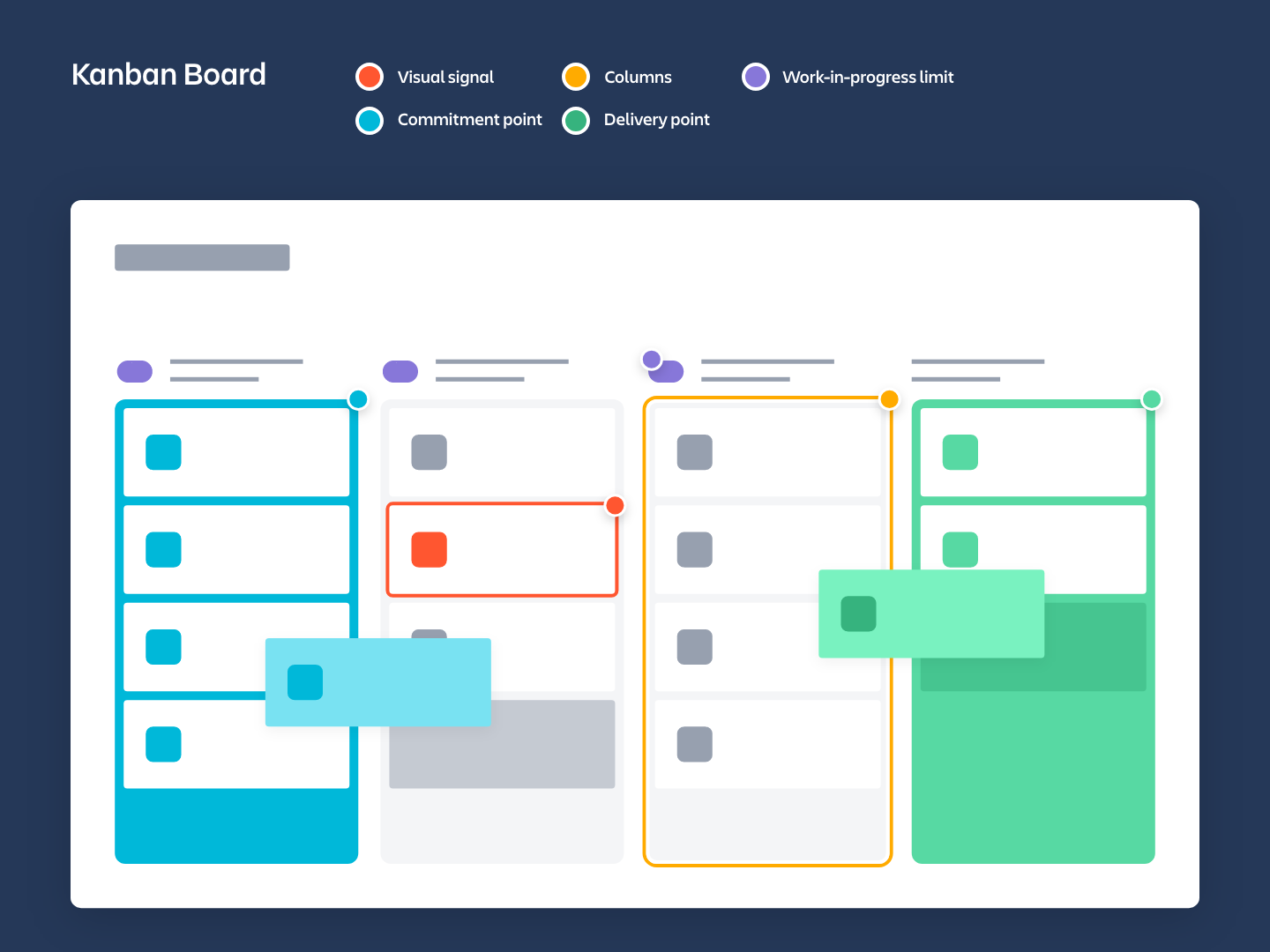
Firstly, visual signals offer teammates and stakeholders information for understanding what the team is working on. Secondly, a workflow consists of columns with different status of work such as ”To do”, “Pending”, “Processing” and “Done”.Thirdly, work-in-progress limits the maximum number of Kanban cards to offer an affordable workload to the team. Fourthly, a commitment point is reached when a task goes to specific stages and is taken by a team member. Fifthly, the delivery point is the end of a Kanban workflow.

Source: atlassian- What is a Kanban board?[[2]](#footnote-1)



**Figures 2.1: Physical boards Figures 2.2 : Digital boards**

Source: atlassian- What is a Kanban board?[[3]](#footnote-2)



**Figures 2.3: Kanban boards including (visual signals, columns, work-in-progress limits, a commitment point and a delivery point)**

A Kanban card displayed on the Kanban board represents a task or an activity. It contains helpful information such as the person in charge, the deadline of the task, the priority , the brief of the task and so on.

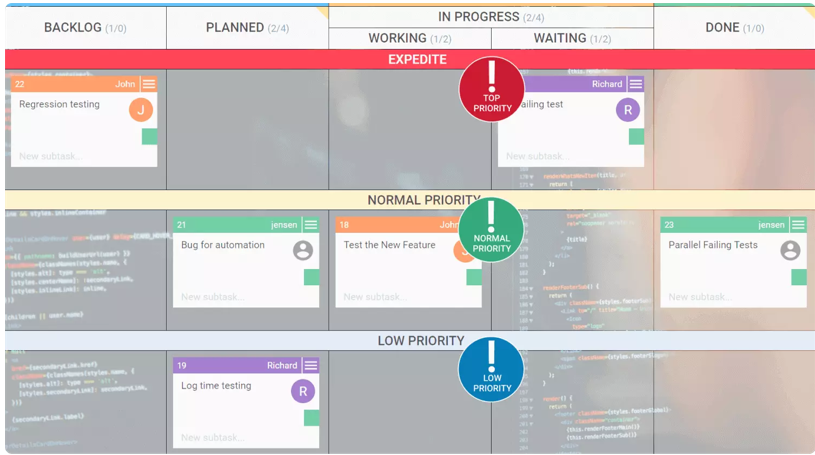
Source: atlassian- What is a Kanban card?[[4]](#footnote-3)

****

**Figures 2.4: Kanban card including basic information**

Kanban is suitable for handling a software project which has tasks and activities with different priority and number of tasks. It is useful for a team to quickly handle small tasks and then output the product quickly. However, It is hard to implement when the software product is complex. For example, lots of work depends on a few specific tasks. Lots of manpower is waiting for the tasks to finish. It may produce a waste of lean which is waiting.

Source: Prioritizing Tasks with Kanban[[5]](#footnote-4)



**Figures 2.5 Different Priority of tasks in the Kanban**

Kanban has two specific roles for achieving the aims. It is service request manager and service delivery manager. The main responsibilities of a service request manager is obtaining and understanding software product requirements or ordering capital goods. Usually, the person having the well communicated skill or having a related job is assigned to take this role such as product manager, project manager or service manager. Moreover, the main responsibilities of a service delivery manager is delivering the software product to clients and holding the Kanban meeting. They are the two specific roles of the Kanban.

**Section 2.2 Benefits and Difficulties for Implementing Kanban**

Benefits of Kanban

Firstly, Kanban can improve the workflow of software development. A Kanban board with cards on board can show tasks which are waiting to be done, tasks which are in progress, tasks which are waiting for review and tasks which are finished. Transparency is achieved by visualizing tasks and allowing team members to observe the stages of a given task. Project managers and team members can allocate or select tasks according to workers’ interest and capability. An employee who is good at paperwork can select to do documentation work or writing reports. Given two team members who have the same abilities, the worker who is interested in doing a task always delivers better results and in a shorter time frame. Implementing Kanban is beneficial for both project manager and project member: project manager can allocate and track the work progress of a member and project members can understand their tasks at hand easily.

Secondly, Kanban can create a sense of belonging to a team. Allocating tasks may not merely be determined by the decision of the project leader. As every pending task is clearly shown, team members can try to apply for the task they wish. Team collaboration is promoted through discussion, deal or coming to an agreement with other teammates. A friendly working environment creates a sense of belonging for each team member. If a teammate gets an unfamiliar task, a well established team can provide support through giving guidance or education because such teams have knowledge and experience from completing projects. A cohesive team can encourage people to learn and share experience with each other and team members are less likely to resign because of the working environment. A project as a team work requires effort from everyone. Combining the above reasons, an experienced team with good relationships will enhance the speed to complete the project.

A team member having a heavy workload puts the project in danger as it may cause a project delay. It signals poor project management skills for task management and time management. With the Kanban, a team member’s workload will be limited to scheduled activities determined by importance and other reasons such as dependency between tasks.

For a good project management practice, focusing on one task only can avoid multitasking and switching between tasks. In the workplace, disturbance by face-to-face communication is common. An employee needs to recall the task progress immediately when the team leader comes to their desk, which may negatively impact if they are thinking of the solution of a task. Although asking for progress through email can be a better alternative, Kanban frees team leaders from reading meeting minutes and instead understands the concurrent activities from the project team by reading the board.

Prioritization can be achieved by ranking the essential task to the top of the lists, this helps the team to understand which task needs to be done first, especially when the time is limited. To satisfy this requirement, a team must discuss each task during an internal meeting. In this meeting, the team has to differentiate jobs by importance, clarify the relations between different tasks by using methods like drawing diagrams to figure out dependency and finally select tasks to do. Such activities can avoid waste of effort. Software development is complex as it involves additional efforts such as creating support documents, like user manuals and drawing diagrams to illustrate a model. Software testing is another essential activity. For sectors like accounting, housing and aerospace, any software faults may cause financial disasters or even casualties. Kanban can redirect efforts to these essential tasks .

The software quality can be enhanced from this model. In the early stage, tasks are carefully selected before starting to work. In the later stage, for software development, if using Kanban correctly, each task should be reviewed by a reviewer before passing the exit requirements. Software inspection with Kanban can serve the purpose of quality assurance. By creating a list of essential tasks and finishing accordingly, time can be saved and more time can be allocated to testing and creating more non-essential functions to serve clients' actual needs. If alpha testing is done and everything goes well, the software can be scheduled for beta testing and scheduled release to the market. Time is a valuable resource and any postponement of the release may impact the sales of the software or bring inconvenience to the client. When a project is done, the workforce can be redeployed to other projects. If the project does not go according to plan, Kanban can help reduce the potential lost by ensuring efforts are only spent on the most important tasks.

If the worker is having difficulty when handling a task, the card will be held in the “In progress” list. The project manager may need to consider which factors are preventing the task from being completed by studying past projects or consulting other members. Moreover, as the tasks are clearly listed on the board, team members are motivated by comparing work done with peers.

Difficulties of Implementing Kanban

A team may find difficulties while setting up or adopting Kanban in their workflow. Using Kanban requires division of tasks into smaller sub-tasks. Kanban requires constant updates according to the status of a task. If the development team is unfamiliar with the usage of Kanban, employees may forget to update the board. As a result, some tasks may become out of sync with the actual task status, preventing dependent tasks from starting. Furthermore, subsequent action such as reviewing the task results may be delayed. As a vicious circle, other activities such as meeting, developing and reviewing shall also be delayed and the project requires more resources for manpower, money and time.

If the project is large, some details may be overlooked. A small board may look too complex and massy and Kanban cards on the board may not include every detail of the task, such as the purpose, the persons in charge, or how the task will be done. Reviewers may be confused. Sometimes, a task may require time efforts, such as adopting peer review. A board or small screen by using software tools may make all the information difficult to read or to understand. Without understanding the task completely, the result may require revision and the result quality would be bad.

Customers may find Kanban is conservative and not inclusive, as the team is supposed to finish tasks which are clearly stated in the board. However, some requirements cannot be stated black and white as customers have their own habits and involve corporate culture. All such requirements are required to learn by communication and observed from the customer's workflow. In this case, the client company should understand the agile practices in the company and try to provide useful information and additional information for the software development as well as actively take part in the development so that important aspects of the software are included in the development process.

For a company newly adopting Kanban, they may find it difficult to use and integrate it into their workflow. Without clear guidance, the team may not understand how to break down the project into many different sub-tasks and how to organize and allocate the tasks to the teammates. As a result, the process may take longer than expected and the management team may feel disappointed for the time it takes. Management team may recommend other agile methodology over Kanban to the development team. To avoid this problem, project managers should manage their expectations by stating that adoption of a new model requires time and effort, and let everyone understand the benefits of adopting this method, and potential problems which may occur. From a project management point of view, a contingency plan is recommended for risk management in order to make sure the project can still be finished on time.

To reduce the friction of switching to Kanban, the whole development team including teammates and a team leader should familiarize the general process of Kanban. For additional support, the company may hire consultants from an external well-practiced party to provide suggestions and supervise the implementation. Frequent revision of the teams' Kanban practice in the first few months, for a shorter feedback loop, ensures some standard is created within the team and provides feedback for them to improve. Gradually, the team will master the skill of applying Kanban.

**Section 2.3 Comparing Kanban with Other Models**

Kanban was originally developed as a process model for car manufacturing in Toyota Motor Corporation. Lean software development evolved from the core of the Toyota Production System which is the foundation of Kanban, will be explained in this section. On top of that, we will compare Kanban with other software models.

The lean manufacturing system is evolved through trial and error and to be developed in the fastest and the most efficient way. The system has two concepts - jidoka and just-in-time[[6]](#footnote-5). For jidoka, the idea is to prevent defective products being produced. If any fault is spotted during the production process, the development process will immediately be paused and the product design or development process will be revised. For just-in-time, the idea is to only produce what the product requires. As a result, all waste is eliminated and the cost is saved. These characteristics can easily be found from Kanban. Kanban cards should be clearly reviewed and then being put into the Kanban board for later development. If any difficulties during the development, the card shall remain the step. Engineers shall check and fix the problem to make the production process run smoothly.

**2.3.1 Comparison with Other Agile Development Frameworks**

Source: agilemanifesto[[7]](#footnote-6)

Table: Manifesto for Agile Software Development

| The Agile Manifesto  We are uncovering better ways of developing software by doing it and helping others do it.  Through this work we have come to 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 there is value in the items on the right, we value the items on the left more. |
| --- |

Agile was developed from iterative and incremental software development in the 1990s. As software design is becoming complex, from the agile manifesto, some aspects are emphasized to be more important than the others. The method is considered to be lightweight contrary to the traditional model which required supervision and plans. Some famous agile methods include Scrum, Extreme programming(XP), Feature Driven Development (FDD) and Rapid Application Development(RAD).

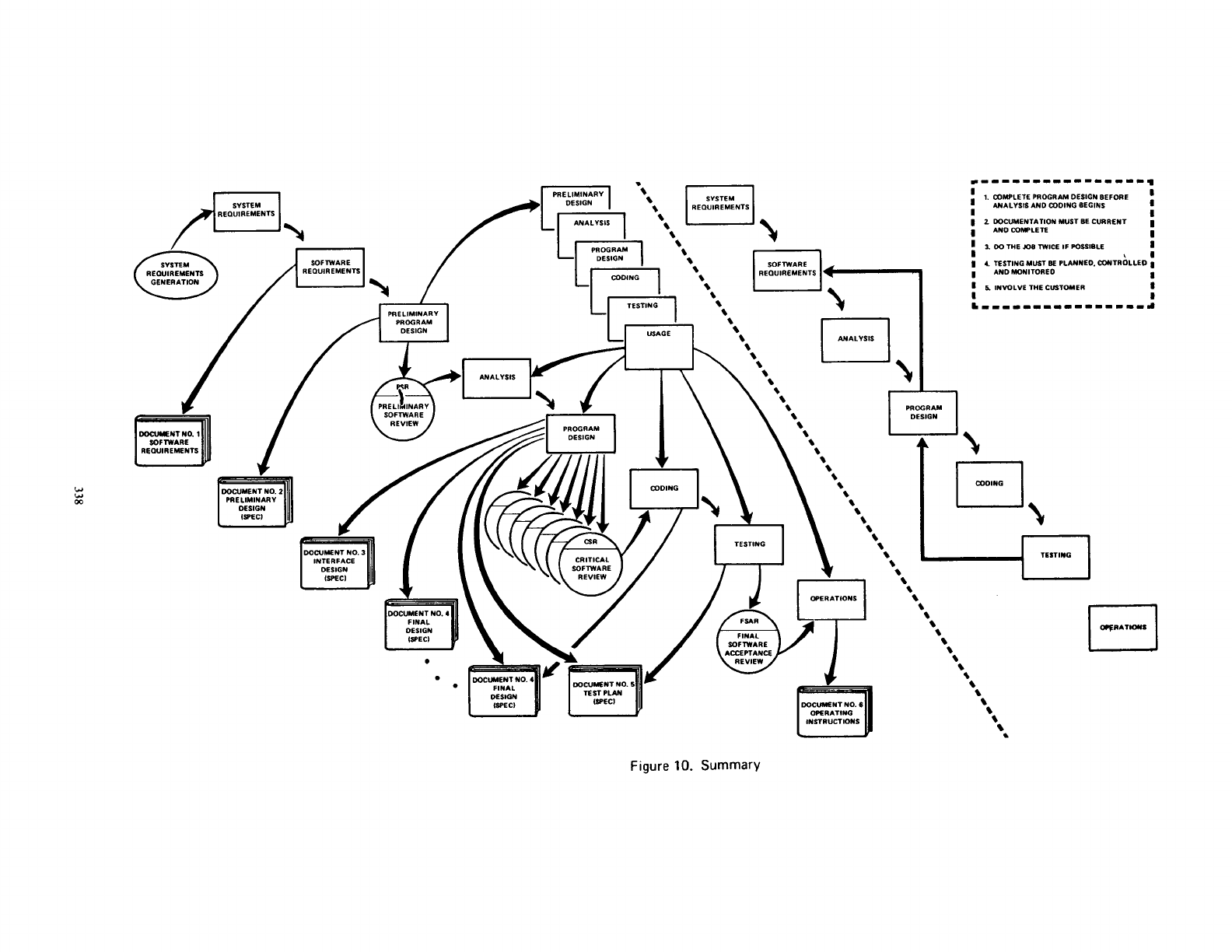
From the first principle, agile encourages collaboration for the whole team or even the whole department rather than only rely on tools determined in the beginning. Following solely on the plan may not produce results which can satisfy customers’ changing requirements in the end. For the second principle, the product is more important than the documentation to explain the design or the usage of the product. For the third principle, a contract has fixed terms signed in the beginning. For a good product, the product should satisfy customers’ needs not only in written paper, but their actual needs. The development team should be able to observe from customers’ workflow to see if there is any potential design. For the last principle, software development may require changes of plan for many reasons, such as usage in technology and human resources allocation. The agile method should be insisted on change rather than stick closely on the original plan.

Comparing other Agile development frameworks to Kanban, their use case scenarios are different. The other is more focused on the changing needs of the software. For a time period, a new version with changes shall be delivered to the customers for testing and approval. After collecting feedback, the development team shall fix or modify program codes based on collected comments. As for Kanban, such a time constraint generally did not exist. Kanban is more suitable for predictable product development as customers’ participation is not the focus during the development phase. Customers’ participation only occurs when doing product evaluation. Some companies may therefore improve Kanban by integrating with agile methods to adopt incremental deliveries and increase the times to collect customers feedback.

Apart from customers’ requirements, there are other aspects to make the two methods distinct. For Scrum, an additional role called scrum master other than the team leader may supervise the implementation of the method. For Kanban, such roles do not exist and the tasks allocation are only based on team culture. Finally, the way of showing information may be different between the two practices. For Kanban, everyone can check on the board for any information of the progress. For agile development, a stand up meeting relying on verbal communication is preferred.

**2.3.2 Comparison with Waterfall Model**

Source: Managing the development of large software systems: concepts and techniques[[8]](#footnote-7)



**Figure 2.6 Summary of Winston W. Royce’s Model**

According to IEEE, a waterfall model may include concept phase, requirements phase, design phase, implementation phase, test phase, and installation and checkout phase. Some phases may be overlapped but with little iteration[[9]](#footnote-8). Waterfall fall model was originally developed by Winston W. Royce for his experience for large computer system development in the 1970s. At that time, a computer was large and expensive, so he put the focus on the design and analysis. He insists on the importance of documentation and suggests the design and documentation should be reviewed at least twice for improving the quality of software development and software maintenance.

The waterfall model is similar to Kanban in many ways. Royce promotes for the waterfall model that documentation should be written as detailed as possible so that stakeholders like programmers, testers and customers can understand the software comprehensively. For each function, it should be divided into smaller sections for detail elaboration, such as purpose and design. When defects occur during software development , the team should take a step back to identify and solve the issue. If the problem cannot be fixed, the team may go back to the design phase to revise the function.

Kanban has the principle for jidoka that paused the process and revised the design. Tasks are separated into subtasks and a task will be stuck on the board when the task cannot be done. The development teams would have to look and fix the problem and create new Kanban cards if necessary. In this process, the team may try different techniques, if the problem remains exist, the team may have to redesign the function to fix the anomaly. Both methods apply to the idea of divide and conquer and the method is good for identifying and fixing defects.

Kanban is required just in time and is more similar to the agile method. For the documentation, it may not require as strictly as the beginning of the product development in the waterfall model. For the waterfall model, the design is applied to large scale projects and even to the national level. Different departments may be involved in the development so that the documentation should be as detailed as possible. Some information may not directly relate to the software. Information should as regulations, rationale and development of the technology are possibly included in the document. On the contrary, Kanban shall specify tasks to do and only the tasks directly related to the product for delivery. For each task, a specific outcome should be specified so that the outcome of the activity can be evaluated.

**Section 3. SQA/SQM activities/tasks**

**Section 3.1 SQA/SQM Standards from IEEE 12207-2017[[10]](#footnote-9) with Kanban**

In IEEE 12207-2017, section 6.2.5 provides information for the software management process and section 6.3.8 provides information related to the quality assurance process. More information can be found on the document.

Software quality management is to ensure the process of development meets the objective and make customers satisfied with the results. The results must be clearly defined and collected. To ensure the quality, the core idea is to set up some criteria and evaluate the implementation. The results should be taken into account for future development and improvement. The activities include planning, evaluation, correction and prevention.

For the planning, firstly, the team should gather all the information about regulations from the firm or the law. Then, the project requirements should be clearly gathered and analyzed. The guide to define how to achieve each requirement should be done. Secondly, roles with authority should be able to monitor and control to ensure the progress sticks to the requirements. Relevant information should be provided to the regulators. The regulator should have a scientific method to measure the outcome. Kanban is helping to follow the progress in step one. Resources from many sources can be marked sequentially. For the second step, a Kanban card may contain information for the worker who is doing a task and the card name implies how the task will be done. For gathering resources, the card may provide a remark to indicate the resources or the contact information for the person who is familiar with the matter.

For the evaluation, all the regulations should be followed, listed requirements should be satisfied and customers agreed with the results. The evaluation process has to be done in iteration for a specific time period. Applying scheduling to the board may ensure the review could be done repeatedly and situations can be observed.

For correction and prevention, evaluation results should be accessed and applied for improving the product and the process. For unsatisfactory results or time out , the development team has to follow suggestions provided by regulators and reviewers. Prevention happens when the observers find that the development is going in the wrong direction. For example, a team may focus on a minor task and a milestone is approaching. The prevention action is to suggest important things first. Kanban board lists out all the relevant tasks to let a team think about the priority and focus on relevant items for customer satisfaction. The board may remind the reviewer to inform relevant stakeholders to take follow up actions. Only when review is completely done, the card with the evaluation task shall be removed from the board.

Software quality assurance is a process to validate the project sticking to the software quality management process so that confidence can build up for the software quality following policies, defined procedures and requirements. The outcome can be divided into two parts. The first part of the procedure is the same as software quality management. The steps are planning for the procedures, defining rules to ensure the procedures are properly executed, checking on the rules to confirm that the progress is sticking to the rules and informing relevant stakeholders. For remarks that software quality assurance is generally conducted by personnels outside the development organization. Kanban record keeping in digital format may be appropriate to be assessed, but the board should not modify or mix with the activities from the assurance team. For the assurance team, given that suppliers from different companies and many configuration items are required to be accessed, Kanban remains a useful tool to record all the status of activities.

As the role of Kanban is already discussed for part one, this part will be focused on the remaining outcomes of assurance. The additional activities are managing quality assurance records and reports. The reports should include information of encountered problems and incidents.

For quality assurance report management, report creation, classification, storage and maintenance ,and distribution are the activities to be done. For each anomaly discovered during investigation and identified to endanger the fulfillment of the requirements are required to sign up for a record and finally included in the report. For all the records, each of them should be classified into a class by its type. The types may include the anomaly related to the product, process or service provided by suppliers. Apart from the type, some information is confidential and only authorized persons are allowed to access, so access control shall be marked for proper storage and maintenance. Kanban cards are suitable for such classification tasks with a clear process to affirm that no record is suffering from mishandling.

Assurance team should inform stakeholders for follow-up actions and update the record once a while. Each record may contain reasons and treatments. Errors are possibly found from past records. All the treatments could be listed and evaluated sequentially. Each issue shall be fixed by severity. Regardless of the treatment's success or failure, the results shall be marked in the record. If the problem is resolved, the record status can change to close and the record may be helpful in the future when similar events happen. Given that many steps are included in the assurance process, it may be too complicated for Kanban as a card on Kanban board is not supposed to take off unless the process is finished. In this case, a to-do list for staff used would be a better alternative. Kanban may follow up the status for each record and it remains useful to prevent missing tasks from the workflow.

**Section 3.2 SQM activities and practices with Kanban**

Assumption of all companies has two teams which are software development team and the quality assurance team.

Kanban software process model offers a quality assurance team a user-friendly environment to assure software quality. Adding the “Quality Assurance” Phase to the Kanban board presents what the quality assurance team needs to do. The quality assurance part can be divided into 3 columns which are planning, Coding and Testing and Review.

**Planning phase**

As a quality assurance team, they need to do the test plan of the application function. The quality assurance members may find the problem of the function before planning the test plan. If the problem affects the integration of the parts or affects the up-comming processes such as logic error, they can make the high priority Kanban card. With a high priority Kanban card, the development software team needs to handle the bug before the quality assurance team does the integration testing or system testing . Without the high priority card, the quality assurance team can do the testing of the product.

If quality assurance members have a doubt of the function, they can communicate with the person in charge by commenting on the Kanban card in order to clarify the function. After the communication, they can do the test plan with a detailed explanation of the function. With a detailed requirement of the function, quality assurance members can create more test cases to test the programmes, improving the product quality as a result.

With a Kanban board, the quality assurance team and the project team can track the problem or questions using the same system. By commenting the issue inside the card, more details be provided to the development team in order to fix the issue in a timely manner. Therefore, It can solve the critical problem first and ensure the software product quality.

**Coding and testing phase**

After finishing the test plan, the quality assurance member can code the test cases with a tool such as junit. When they finish the coding and pass all the test cases with the test plan to ensure that the function can fulfill the clients’ requirement, quality assurance members can submit the testing code. The Kanban card’s status will be updated and the Kanban card will be moved to the next column which is the review phase.

**Review phase**

When the Kanban card reaches the review column, a quality assurance member will be assigned to take the review tasks and begin the code review process. However, the Kanban card should not be reviewed by the one who worked on it before, as it is easier to discover the potential blind spot, missing test case, or bugs by other members, enhancing the product quality as a result

When the member reviews the code and then discovers that the test case cannot cover all the functions of the application they are in charge of, they will make the comment to the Kanban card in order to remind the test case programmer of what cases they need to add. If the reviewer discovers the problem, the viewer will comment on the Kanban card and then the Kanban card will be moved to the Coding and Testing Phase. It can ensure that the test case can fulfill the requirement of the product. With Kanban, this software quality management activities can be easily implemented.

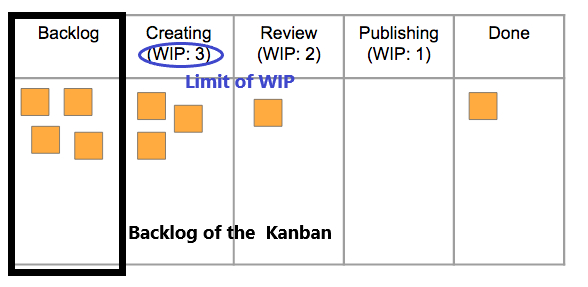
**Section 3.3 Effectiveness in SQM with Kanban**

To begin with, Kanban is a key to identify the problem quickly and resolve it faster. Kanban provides a Kanban board and cards to a software quality team which can visualize project progress and make the team easily understand which task is waiting, processing, or finishing. Commenting on the Kanban card, any team member can easily communicate with the coworker when they counter a problem. It can reduce the time spent on meetings. Team members can comment on the tasks and suggest how to enhance the software product with a real time commenting. It is convenient for software quality teams to handle the software quality management. Therefore, Kanban offers effectiveness in SQM with a visual workflow.

Moreover, having the limit of work in process can help project managers to improve the workforce plan. Providing extra manpower to assist for the process which encounters problems may reduce delivery time of the software product and then improve the software development cycle time. It can avoid that a person is overloaded and then cannot submit the product on time. Also, having enough time to finish a task is a suitable way to offer high quality outcomes. Kanban focuses on finishing the tasks you are working on but not getting more tasks. It can counter the problem that the staff cannot meet the deadline or deliver low quality products caused by over workload. Limiting work in process can ensure the product quality is not affected by over-committing to work and also enhance the workforce planning with the visualization of workflow.

With the Kanban, the project manager can create a suitable plan for distribution of tasks to produce a high quality product. To prevent disrupting scheduled work of each staff member, the project manager will not allot new tasks to their teammate when they are working on their own tasks. As it may overload the staff member and affect the product quality. Using Kanban can improve planning to lead to faster cycle times. For example, it can reduce the waiting time of work. Therefore, Using Kanban is useful in software quality management.

Source: agilesherpas[[11]](#footnote-10)



**Figures 3.1 Backlog of the Kanban process and Limit work in process**

**Section 3.4 How Kanban adheres to the general Agile Manifesto**

Kanban has two main concepts which are change management principles and service delivery principles. Having these principles are the key to achieve the general agile principles and practices.

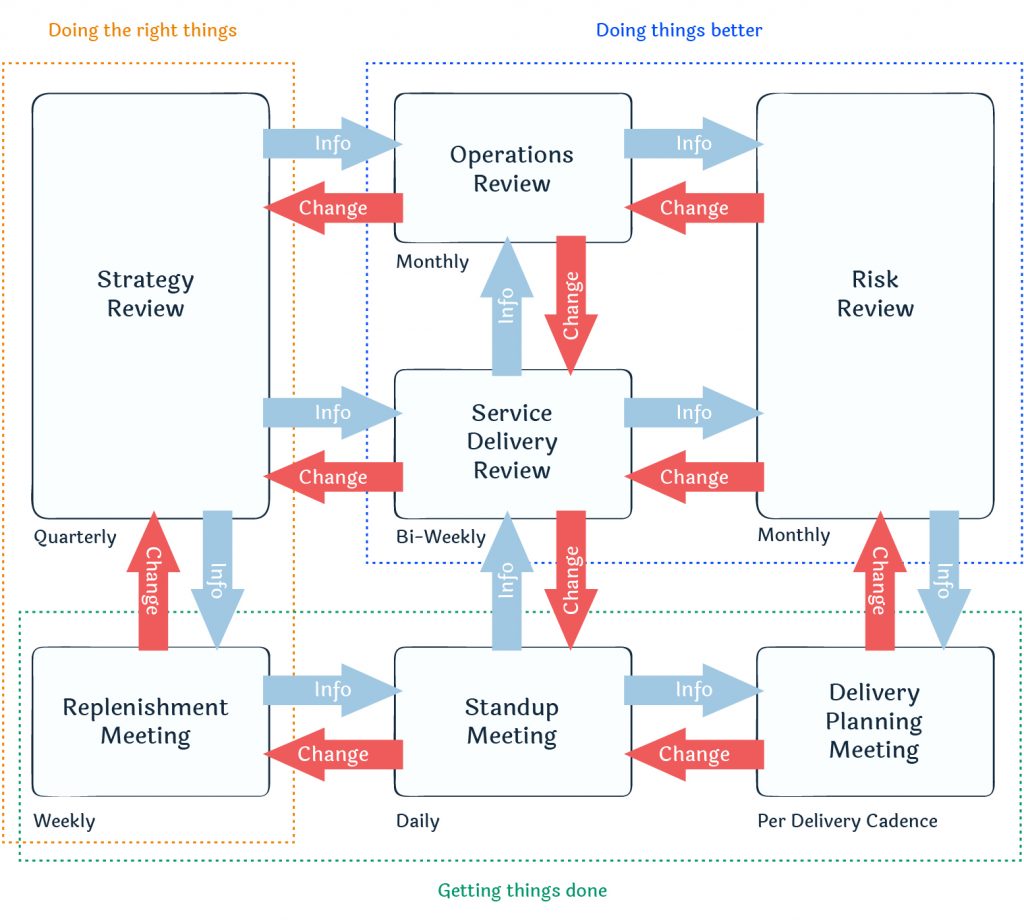
Change management principle has few principles which are: start with what you are doing now, agree to pursue improvement through evolutionary change,encourage acts of leadership at every level[[12]](#footnote-11). Kanban can adhere to these general agile principles, which will be explained below.

Firstly, starting with what you are doing now can reduce the disruption when the team is working on their own part and make them focus on the task they need to handle now. Kanban visualizes the workflow in order to clarify the process and make the team easily understand what they need to do first. The team members can start their work faster when the project starts with the Kanban card they are assigned. It can reduce the waiting time and make faster cycle times without the noise of understanding the process. It makes Kanban achieve the agile principles which are delivering working software frequently[[13]](#footnote-12) and offer team the affordable amount of tasks to maximize efficiency[[14]](#footnote-13).

Secondly, Kanban keeps offering the team the process of improvement. The project manager can measure the effectiveness with the Kanban such as the number of the product they submitted during a period, the number of complaints from clients, the cases in which the product cannot be submitted on time and so on. Therefore, the project manager can take actions to improve collaboratively such as assigning work according to team member’s strengths, offer team members the opportunity to raise their opinions in order to improve the process and so on. It is useful for the project manager to remove the ineffective action in the process and make useful evolution of the process to improve the process change. It can adhere to agility principles which keep enhancing the agility with continuous focus on improving the performance of the workflow.

Thirdly, Kanban provides all teammates an opportunity to master the soft skills such as communication skill, organization skill and leadership. Each teammate will be allotted tasks. Therefore, they need to handle the task and communicate with the coworker if they have a problem about the process or other teammates have questions to ask them. They can have a choice to lead other people. By practicing their soft skills, the team can be worked effectively together and also they may discover useful skills which can enhance the effectiveness of the process. Using the Kanban comment function, the software development team can also communicate with the business team to clarify the needs of the customer. According to the meeting of the Kanban, all the team members' work will be reviewed in order to ensure the software quality work.

Source: getnave[[15]](#footnote-14)

.

**Figures 3.2 Kanban meeting**

Therefore, Kanban offers lots of opportunities to master the soft skills to enhance team work. It can adhere to agility principles which let business people and developers must work together daily throughout the project[[16]](#footnote-15) and exchange information about the project with an efficient face-to-face conversation.

Additionally, service delivery principles contain few principles which are understanding and focusing on customer needs and expectations, managing the work[[17]](#footnote-16). These principles cause the team to offer a good quality of product using the Kanban agile software development process.

To begin with, understanding and focusing on customer needs and expectations is an element of creating a high quality software product that meets client's requirements. Kanban offers a fixable way to handle client requirements. When the marketing team gets a new request from the client, they can create a Kanban card or add a new comment in the related cards to alert the development team to add new functions or update the function. It enhances the communication between the marketing team and the software development team to fulfill the clients needs. It can adhere to agility principles which are: having a highest priority to fulfill the client's need, and welcome changing requirements for the customer’s competitive advantage[[18]](#footnote-17).

Moreover, Kanban offers team detailed information of a project for managing their work in order to assure the product quality. By having a visualized workflow and the limiting work in progress, the teammate can self organize their own work, follow the schedule and enhance the task with comments. In addition, the manager can easily measure the process of the product and discover bottlenecks using the Kanban board. For example, viewing the priorities of the task, the manager can make the decision to enhance the performance of the process. When a task with high priorities is processing without enough manpower, the manager can offer them aid with an experienced colleague to finish it first. Therefore, it is helpful for managers to provide aid to other colleagues who encounter problems. It can achieve agility principles which offer the team a friendly environment in order to support them and finish the task with high quality outcome and enhance the process with adjusting behavior accordingly. [[19]](#footnote-18)

**Section 4. Presentation scripts (and slides in a separate file)**

**Part 1 of powerpoint slides (number 1 to 23)**

[Start from Page 1]

Au Ka Fu:

Good Morning everyone. We are group 2 and the software process model we are going to present today is Kanban. I am the project leader, Kidd, and following our team members will introduce themselves.

Chung Ho Kin:

I am Ken Chung.

Lam Ka Yik:

I am Ken Lam.

Chan Shing Hei:

I am Rena Chan.

Leung Man Yuen:

I am Leung.

[Move to Page 2]

Au Ka Fu:

Here is our agenda of the presentation today. First, we will talk about the basic concepts of Software Quality. Then, we’ll introduce the Kanban model. After that, we will explore the Software Quality about Kanban. Finally, we will share information about software tool and give a conclusion.

[Move to Page 3]

Au Ka Fu:

Ok, let’s start with the Software Quality first.

[Move to Page 4]

Au Ka Fu:

There are some basic concepts that we have to know about Software Quality. SQA, Software Quality Assurance, which is a process that ensures the quality of a software. During this process, the project team will check the developed software periodically. It help reduce the risk of development process and enhance the software quality within time and budget.

[Move to Page 5]

Au Ka Fu:

Software Quality Management, similar to the SQA, is an aspect that guarantee the software quality and meet the project requirements. It is also a process throughout the whole software life cycle.

[Move to Page 6]

Au Ka Fu:

As we’ve just mention about software life cycle, so what is software life cycle? Software life cycle, also known as software development life cycle. It describes the process of planning designing and developing a software. Its detail can be defined by different people, but commonly it includes 6 stages: Requirement gathering and analysis, design, implementation, testing, deployment, and maintenance. There are many software quality practices can use in Kanban and we will talk about it in the later section.

[Move to Page 7]

Au Ka Fu:

Besides the SQA and SQM, there are many other concepts about Software Quality, for example, Continuous Integration, which means the automated task that ensure the integrity of code changes. Continuous Delivery, similar to Continuous Integration, is also the automated task that package changes to users. Both concept usually consists 3 stages: build, test and deploy, in which developers can define what kind of services will be used in these stage.

[Move to Page 8]

Au Ka Fu:

After introducing the concepts of Software quality, let’s move on to talk about Kanban. I will give the simple background of Kanban and then will invite my teammate Ken Chung to continue.

[Move to Page 9]

Au Ka Fu:

Kanban is first introduced in early 1940s in Toyota Motor Corporation. And it came to software development by David J. Anderson in 2004.

[Move to Page 10]

Au Ka Fu:

Kanban is a methodology to handle tasks and processes. It included 2 parts which are lead time & cycle time. Lead time means the task is placed into the product backlog and before delivering the product to the customer. And the cycle time refers the task is started and put into the Kanban Board until the end of the tasks. Kanban always included 2 elements, Kanban boards and Kanban cards. Now, let’s invite Ken Chung to continue to introduce more detailed about Kanban.

[Move to Page 11]

Chung Ho Kin:

Thanks, Kidd. So, after listening to the basic background about Kanban, you may ask “what Kanban can do?”. There are some main objectives in a software development process, Kanban can visualize the project workflow. It can limit the work in progress, continuously improve the project. It can also help the project to focus on the process flow.

[Move to Page 12]

Chung Ho Kin:

The Kanban matches 2 principles, which is change management principle and service delivery principle. To begin with, change management principle can offer a team a new working style. Viewing the Kanban can let the team knows what they need to do. It is helpful to build the teamwork.

Moreover, service delivery principle focus on the needs of the client and the quality of the product. It can help reduce the risk of errors in order to enhance the outcome of the product.

[Move to Page 13]

Chung Ho Kin:

So, how does Kanban work? Kanban is always composed by 2 components, which are Kanban boards and Kanban cards.

[Move to Page 14]

Chung Ho Kin:

First, let’s talk about the Kanban boards. A Kanban board contains 5 elements, visual signals, columns, work-in-progress limits, commitment point and delivery point. It can either be in digital style or physical style. Firstly, visual signals offer teammates and stakeholders information for understanding what the team is working on. Secondly, columns refer to the different status of work. Thirdly, work-in-progress limits the maximum number of Kanban cards. Fourthly, a commitment point is reached when a task goes to specific stages and is taken by a member. Finally, the delivery point is the end of the workflow.

[Move to Page 15]

Chung Ho Kin:

Let’s have a look on the basic Kanban board. It shows us 3 columns on the board, which represents the 3 status of work: “To Do”, “Doing”, “Done” respectively.

[Move to Page 16]

Chung Ho Kin:

And you can see the difference between digital Kanban and physical Kanban here.

[Move to Page 17]

Chung Ho Kin:

Besides the Kanban board, Kanban card is the second elements of a Kanban. It displays on the Kanban board. And each card represents a task or an activity. It contains some information, such as person in charge of a task, the deadline of work, and etc.

[Move to Page 18]

Chung Ho Kin:

Here is an example of Kanban card. We can see different information from this card. And the fields of a Kanban card is also customized, which can decided by different team.

Then I will pass it to Leung and let him talks more about Kanban.

[Move to Page 19]

Leung Man Yuen:

Using Kanban in project has lots of benefits. It has high flexibility to different project. It helps create friendly working environment by distributing the tasks more fairly. It can also prevent the exceeded workloads by the work-in-progress limit. Visualization of the process workflow also increase the productivity and efficiency, by making a prioritization and identifying the workflow problems.

[Move to Page 20]

Leung Man Yuen:

Although applying Kanban will give us lots of benefits, there’s still some difficulties we may encounter during the process. For example, a team which is unfamiliar with the Kanban model may feel difficult to divide the whole process into different tasks. Besides of that, the team members may forget to update the latest work status to the Kanban, and it may leads some delay or errors. And the biggest challenge of Kanban is that, it cannot support a too complex system, since it may include too many tasks.

[Move to Page 21]

Leung Man Yuen:

After talking lots of information about Kanban, we can now apply the Kanban into the SQA, SQM and SDLC easily. For SDLC, Kanban allows developers to visualize the workflow and work parallelly. Also, Kanban is flexible from system to system. It suits the SDLC since SDLC can define different stages by different team.

[Move to Page 22]

Leung Man Yuen:

Besides SDLC, Kanban can also practice the SQ . It helps to evaluate processes by adopting a plan-do-check-act approach. It is also able to adapt the change of process and monitor the whole situation. It is useful when creating, updating or reviewing configuration items. More details about the SQ activities will be introduced by other teammates. After deeply exploring the Kanban model, I believe all of you now have better understanding about Kanban.

[Move to Page 23]

Leung Man Yuen:

Besides the Kanban model, there are many process model that we can use. And the most common and basic one is Waterfall model. To compare these two models, we can find that there are some similarities and difference. First talk about the similarities, for example, both Kanban and Waterfall need to revise the process when there are something go wrong. And it also need to divide some bigger parts into smaller parts. Both may get stuck if some problem cannot fix. But the difference is that, Waterfall is more suitable to the large project scale than Kanban. Also, the documentation of Waterfall needs to be as detailed as possible, not like Kanban.

**Part 2 of powerpoint slides (number 24 to 43 )**

[Move to Page 24]

Leung Man Yuen:

Then, we will explain the SQA/SQM activities and tasks.

[Move to Page 25]

Leung Man Yuen:

First, we talk about Software Quality Management. There are four phases planning, evaluation, correction, and prevention. The team should gather and analyse all information and project requirements for the planning phase. Kanban will record the tasks into the card and show them on the board. For the evaluation phase, the team gathered the functions and started to satisfy the listed requirements, and the customers agreed with the result. The team can follow the card's priority on the board to finish the task. For correction, the team needs to analyse the items that do not achieve the software quality objective. In Kanban, the quality team should review the task, find out the items that do not fulfil the software quality objective and provide the correction and prevention actions like creating reminders, notes on reports, and creating a guideline. Finally, the prevention phase should inform the related stakeholder. To notice the problem and remind the problem to happen again.

[Move to Page 26]

Leung Man Yuen:

Software Quality Assurance is a process to validate the project by sticking to the SQM process. There are two parts to the final product. The first step is similar to that of software quality management. Planning for the processes, setting rules to guarantee that the procedures are correctly performed, checking on the rules to verify that the progress is according to the rules, and alerting key stakeholders are the steps. For example, software quality assurance is usually done by people who are not part of the development team. The board should not change or mix with the actions of the assurance team. Kanban record keeping in digital format may be appropriate to be examined. Given the need to access suppliers from several firms and many configuration items, Kanban remains a valuable tool for the assurance team to keep track of all activity statuses.

Report management, classification, storage and maintenance, and distribution are all actions that must be completed for quality assurance. Each anomaly uncovered during the inquiry and determined to jeopardize the criteria' fulfilment is needed to sign up for a record, which is then included in the report. Each of the records should be categorised into a class based on its category. Anomalies relating to a supplier's product, method, or service are among the possible types. Aside from the kind, certain information is sensitive and should only be accessed by authorized individuals, therefore access control should be noted for appropriate storage and maintenance. Kanban cards are ideal for such categorization jobs since they have a defined mechanism for ensuring that no records are missing.

Then I will invite Ken Lam to explain an example of Kanban use case.

[Move to Page 27]

Lam Ka Yik:

Thanks Leung.

Now, I am giving you an example of how SQM activities and practices with Kanban. There is a company using Kanban that has two teams Software development team and the Quality Assurance team. The cycle can be divided into three columns: planning, coding, testing, and review.

[Move to Page 28]

Lam Ka Yik:

In the planning phase, the Kanban have a board that is used to display all the problems or questions that find by both teams. The board contains a different card that is about the issue and the priority of the tasks. The team member can visualize all the tasks and solve the critical problem with the highest priority first.

In the coding and testing phase, the development team start to code the system and the QA team start coding the testing program for the system according to the card on the board. The manager can view the status of the test case by the Kanban column.

When the Kanban card reaches the review column, a quality assurance team is assigned to take over the review duties and start the code review process. However, the Kanban card should not be examined by the person who worked on it before because it is for other members to find out any blind spots, missing test cases, or problems, improving product quality. When a team member analyzes the code and realizes that the test case doesn't cover all of the functionalities, they will add a note to the Kanban board to remind the test case programmer of the issues that need to be fixed. The QA team will transfer the card to the Coding and Testing Phase.

[Move to Page 29]

Lam Ka Yik:

Let’s talk about the effectiveness in SQM with Kanban. It can has the limit of work in process can help project managers to improve the workforce plan. It can avoid that a person is overloaded and then cannot submit the product on time. Limiting work in the process can ensure the product quality is not affected by over-committing to work and enhance workforce planning with workflow visualization. However, this may cause the backlog like the picture on the ppt, and this is the bottlenecks when using the Kanban.

[Move to Page 30]

Lam Ka Yik:

Let's go to the next topic, software tools. Please welcome next speaker, Rena Chan.

[Move to Page 31]

Chan Shing Hei:

All right. Today, I'm going to introduce three examples of Kanban software tools. The 1st one is Notion, it is a project management and note-taking software released in 2016. The second one is Trello, a web-based Kanban-style list application, which was developed by Atlassian and released in 2011. The third one is Asana. It is also a network and mobile work management platform, which aims to help teams organize, track and manage their work. They all support web and mobile application.

[Move to Page 32]

Chan Shing Hei:

First, Notion with Kanban. Notion provides a workspace that allows users to create different pages with different blocks components such as board, Timeline, Calendar, and Table. The board function can be used as Kanban development, the same as the picture shown in the PowerPoint. You can see the components like columns, cards, and the number of cards in the list.

[Move to Page 33]

Chan Shing Hei:

Inside the Notion Card. It contains different custom attributes like a status label with pink colour under the title, and assigned label under the status label, and a priority label with brown colour under the assigned label. And then, the user can use the free space under the attributes for remarks such as add table, checklist, and timeline.

[Move to Page 34]

Chan Shing Hei:

Notion provides a significant degree of freedom for customizing. It is highly dependent on users’ needs, such as users can use a different combination of block components, checklists, and tables on one page. Also, Notion support keyboard shortcut keys like using the ctrl and ‘n’ key to create a new page, ctrl and ‘p’ to quick open. The function is similar to some software IDEs like Visual Studio Code. However, Notion has limited automation support. Many processes need to process manually, such as the user need to move the card manually and perform the development pipeline manually. Notion is a lack of obvious visual alert on bottlenecks. The user may need to pay extra attention to managing and finding the bottleneck.

[Move to Page 35]

Chan Shing Hei:

Trello also provides different including Tables, timelines, and Calendars. The table uses to perform Kanban development. You may see the example of the Table on the slide.

[Move to Page 36]

Chan Shing Hei:

Here is an example of the Trello Card. It contains assigning members, adding labels, creating the checklist, and leaving comments for the user to edit and record the tasks.

[Move to Page 37]

Chan Shing Hei:

Trello provides complex automation within the software, such as the chaining actions function to move a card to another list while it meets the card's due date. That allows users to allocate less time managing the board. Also, Trello can integrate with different software project tools. For example, Trello will create a new Trello card when a new "issue" is made on GitHub. However, it needs a subscription to acquire additional functionality.

[Move to Page 38]

Chan Shing Hei:

Finally, our last software tool, Asana. It also provides many views, and the board view is mainly for the user to perform the Kanban development. The card can be accessed on the right panel and allow the user to enter tasks details like an assigned member, task dependencies, priority, and subtask.

[Move to Page 39]

Chan Shing Hei:

Asana provides a built-in reporting page that helps the management team evaluate the project progress. The page offers a different chart for the management team, which is easier to understand progress instead of viewing a bunch of tasks with so many cards on the board. Also, the developers can check their tasks on the personal “My Task” page, which means they can focus on their tasks only and start their tasks without finding the tasks from the board. But Asana needs to perform an additional setup on reporting page. It doesn’t have any obvious indication of the task-in-progress, which means the management team must spend more effort on task management.

[Move to Page 40]

Chan Shing Hei:

Here is an example of the reporting page. You may see the chart in the picture. It provides a different chart for the user to understand the progress quickly.

[Move to Page 41 – 42]

Chan Shing Hei:

OK, so we’re coming toward the end now. Let me summarize the key points. We have introduced the software quality. Then we explain the Kanban history and concepts such as its history, the principles of Kanban, which change management principle and service delivery principle, two main components boards and cards. Then explain the Kanban activities and tasks that fulfil SQA, SQM and how it follows the plan-do-check-act cycle. Finally, we suggest some software tools, which are Notion, Trello, and Asana, can perform the Kanban software development process model.

[Move to Page 43]

Chan Shing Hei:

This is the end of our presentation. If you have any questions, feel free to ask.

Thank you for your attention. I hope you enjoy it.

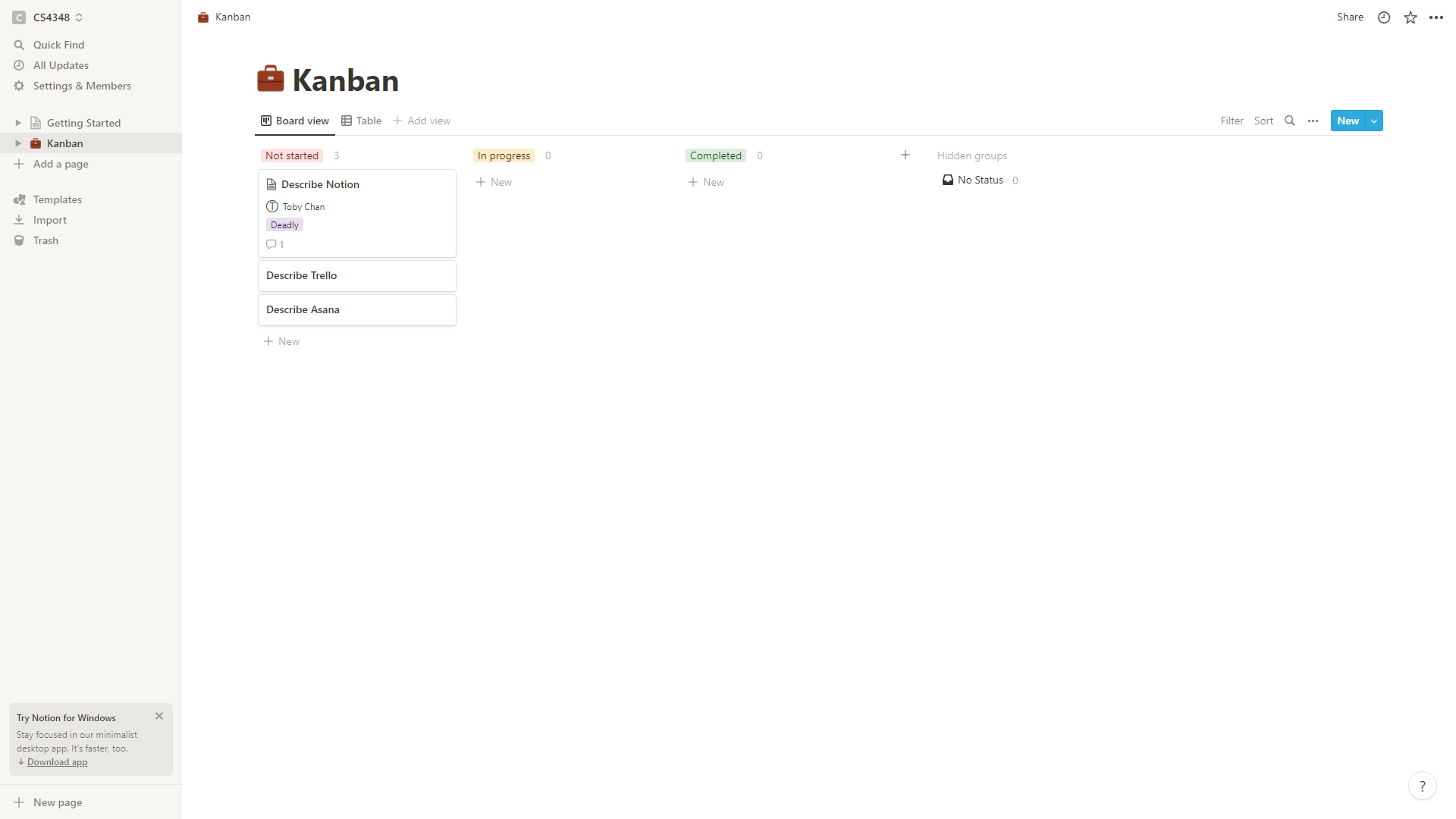
**Section 5. Software tool**

# Notion

## Background

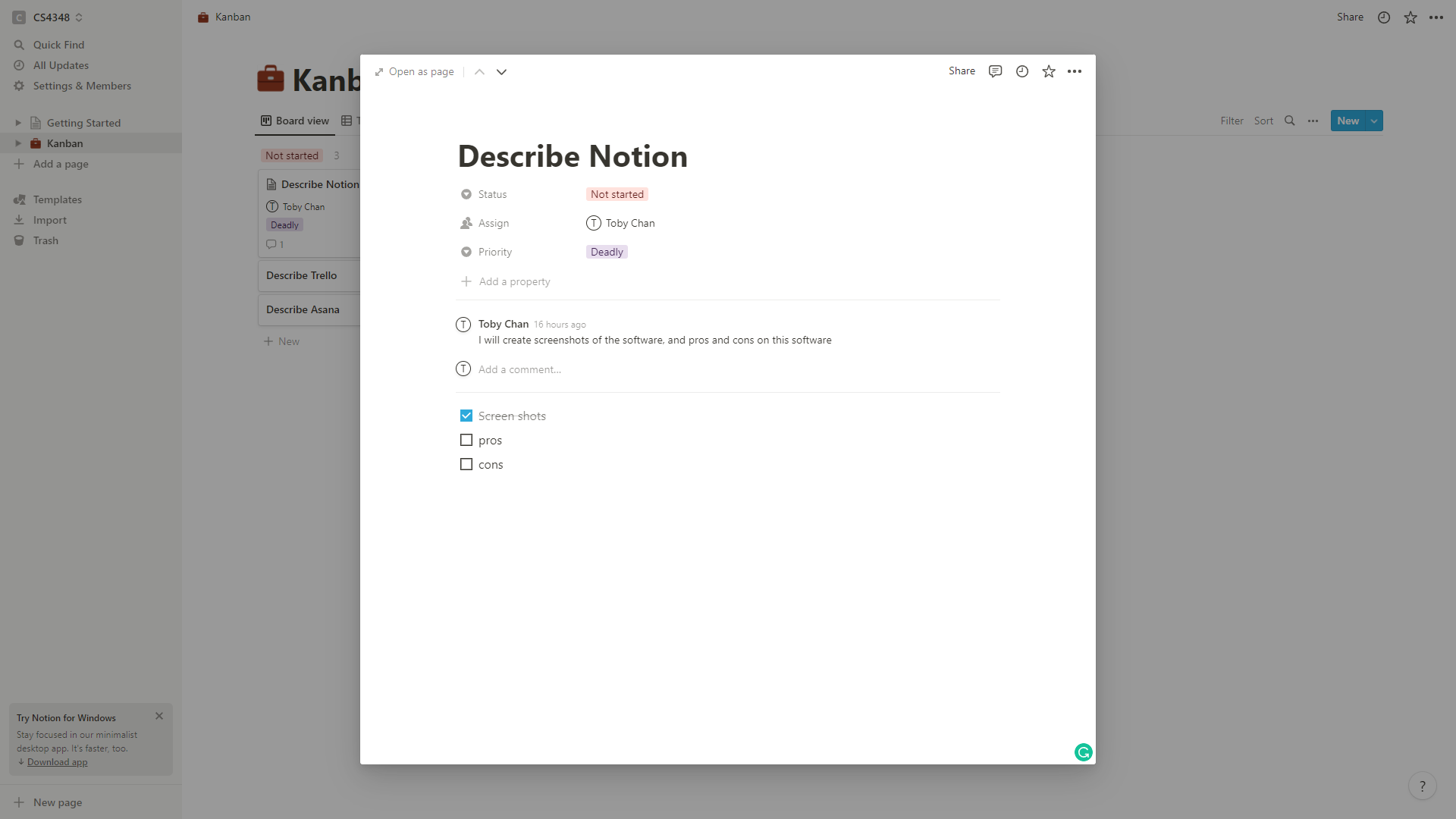
Notion is a online service started back in 2016[[20]](#footnote-19), aiming to create an all-in-one workspace for users like companies or individuals. As the founder Ivan Zhao and Simon Last found that it is a hassle to manage different software (e.g. using gmail and slack to communicate with client/colleges, google document for document related task) just to perform daily office work[[21]](#footnote-20). Therefore, Notion developed this platform that allows users to collaborate within the same workspace, and aims to perform daily tasks without leaving Notion.

## What does it look like

****

**Figures 5.1 Notion Kanban Screenshot**

Notion workspace allows users to create different pages, which can insert different blocks of components. One of the components is a database, providing different views (such as Board, Timeline, Calendar, Table), which its’ board function can be used as Kanban development. The board provides basic visual components such as columns, cards, and number of cards in the list.

****

**Figures 5.2 Notion Card Screenshot**

Above is an example of “card”, featuring different custom attribute (e.g. Status, Assign, Priority), a comment section, and a “free space” for create remark of the task (a checklist in this case)

## Special Feature

Notion provides a large degree of freedom when customizing the Kanban board and cards, for example, users can prevent less important (card) attributes from showing in the Kanban board, or filter cards by combination of (card) attributes, allowing users to quickly understand what they need to know. A relation of different cards can be saved as an attribute of the card, which can be useful when displaying blocking tasks or subtask relationships.

Notion also provides good support for keyboard shortcut keys, some shortcut such as new page (ctrl + n) and quick open (ctrl + p) overlaps with IDEs like Visual Studio Code. This can benefit developers to use Notion intuitively, or let normal users use the software without moving their hands to mouse frequently.

## Cons

Notion currently has limited automation support (both within the software and with 3rd party integration), users may need to move the cards manually (unlike some other software that can trigger workflow by “rules”), and perform the development pipeline manually, this may lead to developers wasting extra time on managing cards.

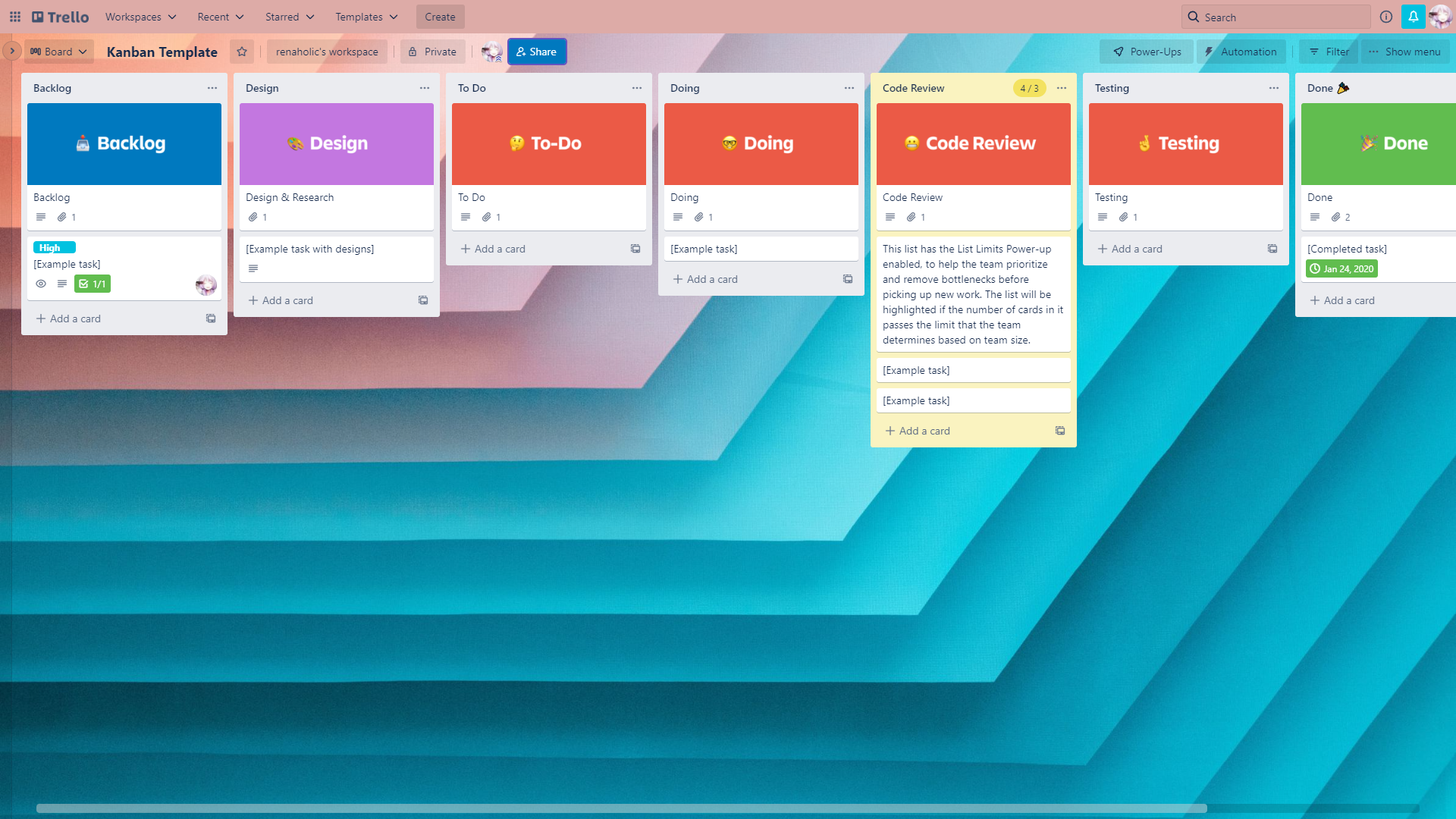
Notion also lack obvious visual alert on bottlenecks (unlike some will turn the background of the list yellow), this may require developer or project manager to pay extra attention on finding out the bottleneck, and limit the number of tasks-in-process themselves.

# Trello

## Background

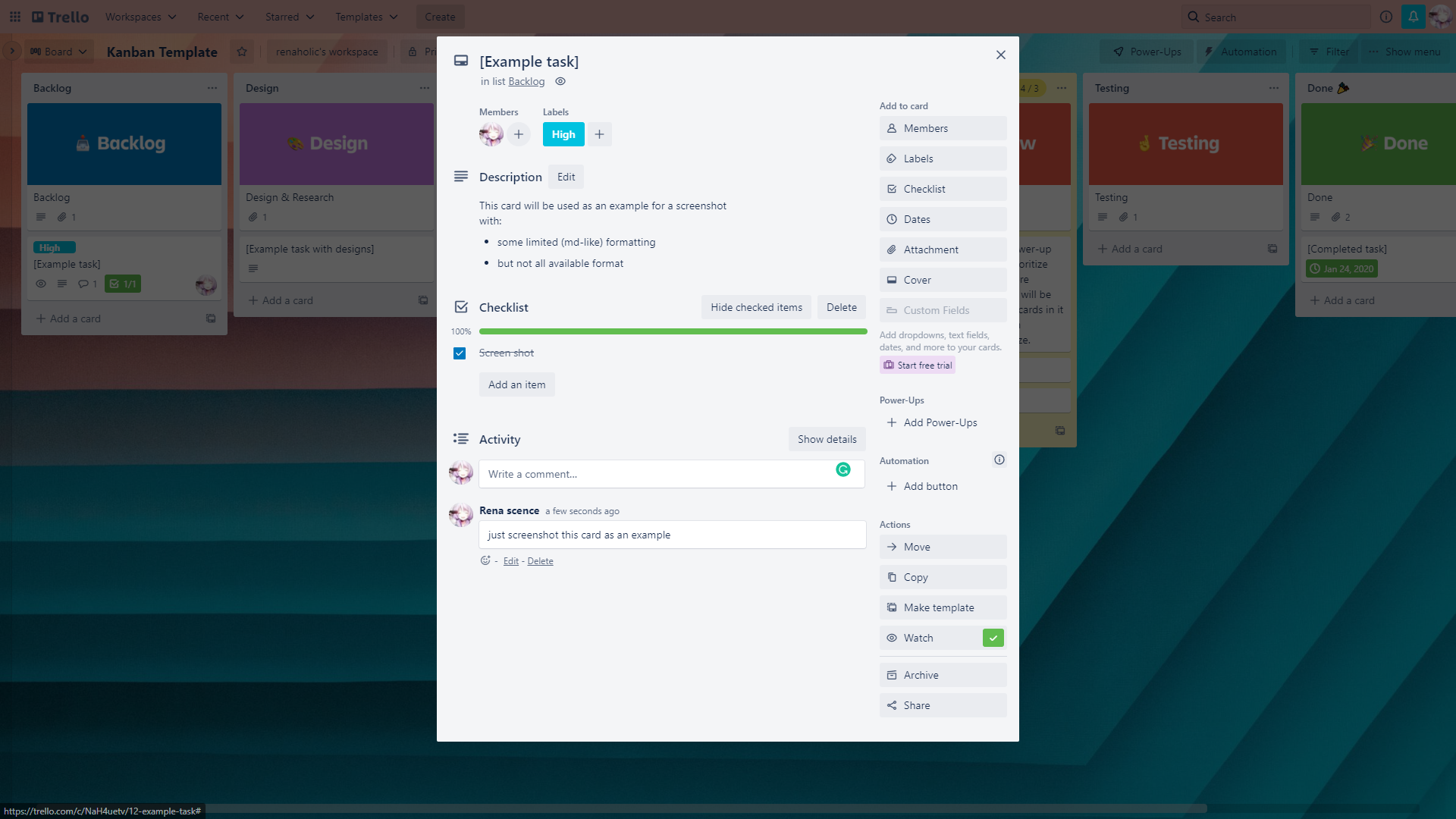
Trello begins as an idea in 2010 by Fog Creek Software[[22]](#footnote-21). With the aim of providing aids in solving high-level planning issue, Trello released their first version in 2011, and currently acquired by Atlassian, a software company specialize in developing service for software developer and project manager. Trello now aims to provide their solution to different team such as product management, technical support, marketing team, etc[[23]](#footnote-22).

## What does it looks like

****

**Figures 5.3 Trello Kanban Screenshot**

Trello’s workspace provides different views (such as Table, Timeline, Calendar, etc), which table can allow us to perform Kanban development. Above is a Kanban template provided by Trello, in which the background of the Code Review column is yellow, indicating potential bottleneck, with which this visual alert is enabled via “power-up” function by default, however, this does not impose a hard limit on the task-in-progress too.

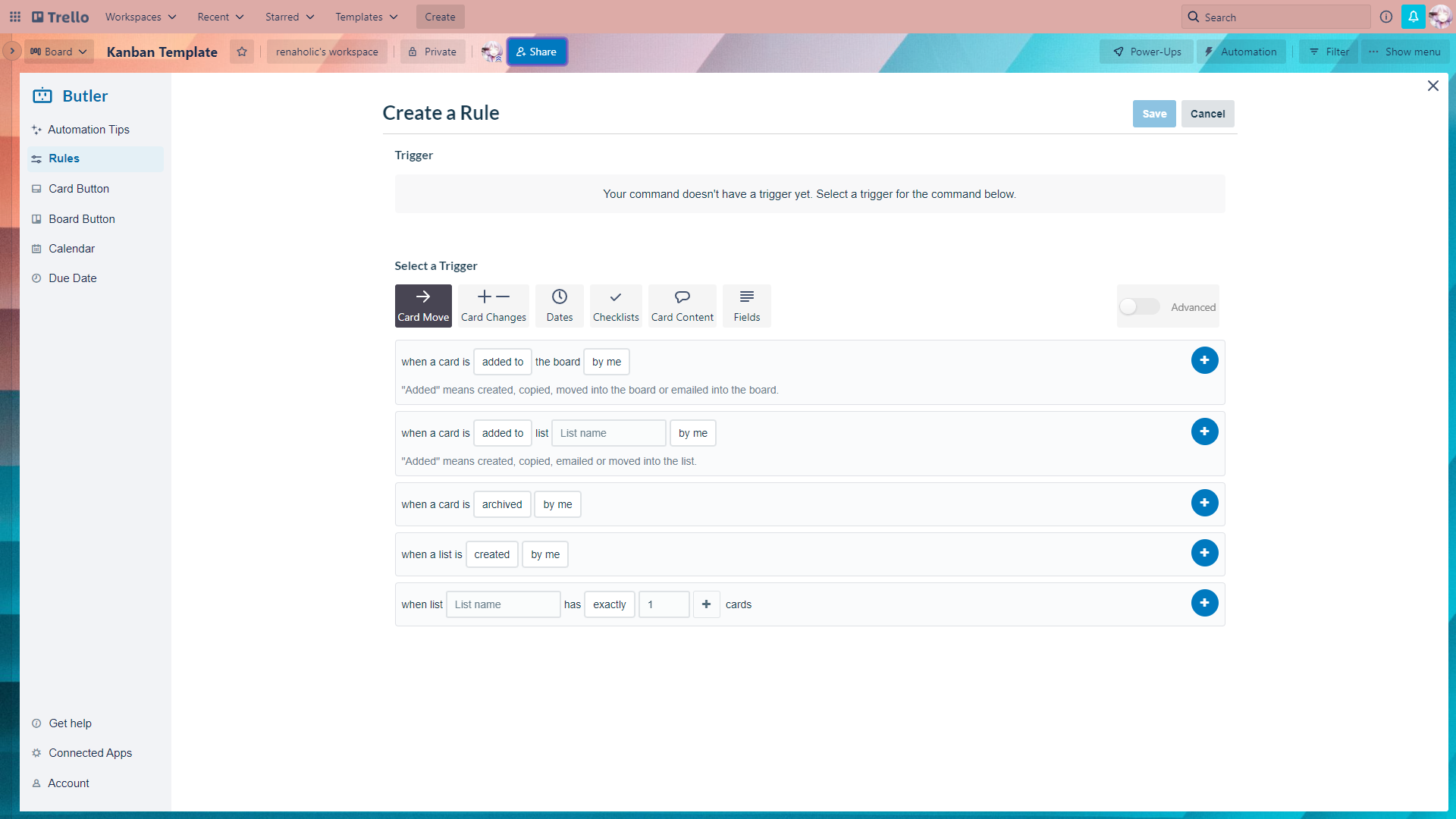
****

**Figures 5.4 Trello Kanban Screenshot**

Above is one of the tasks from the Kanban, allowing users to edit some attributes such as assigning members, adding labels, creating checklists, leaving comments, etc.

## Special Feature

## Trello allow powerful, complex automation within the software (such as triggering workflow by rules, and manually trigger by pressing buttons)

****

**Figures 5.4 Trello “Automation” Screenshot**

Above screenshot is some of the builtin possible automation “trigger”, by creating triggers (such as when a specific user joined a card), and chaining actions (such as moving card to list, and setting due date of the card), this allows the development team allocate less time on managing the boards. By using community add-on (called power-up), the development team can sync cards between boards, perform time tracking, or even create development pipelines with chosen software, such as creating a new card when Github has a new “issue” created, this can reduce development friction by removing repetitive work.

## Cons

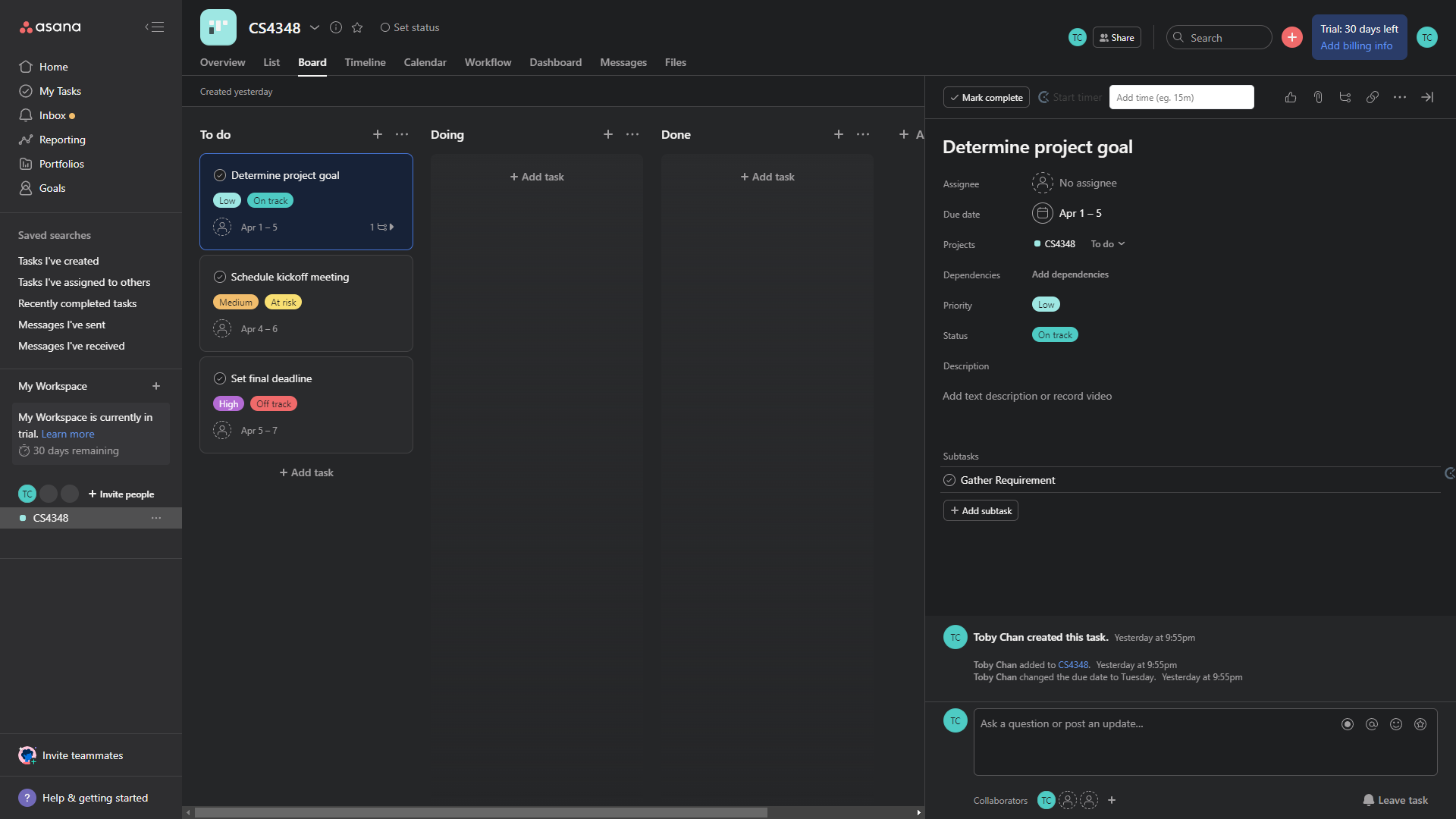
Although Trello provides good Kanban development in free tier, a subscription is required to acquire additional functionality such as different view of the board (such as Timeline, Calendar, Dashboard, etc)

# Asana

## Background

Asana is created by ex-facebook engineering leader, Dustin Moskovitz and Justin Rosenstein[[24]](#footnote-23). They discovered as Facebook grow in size, most employees are spending their working hours on things like meeting, responsing to email, and other things that are “work about work”. Therefore, they created Asana with the aim to empower business by marshalling the work in one connected place, allowing the employee to focus on the work that counts. Currently, there are more than 100,000 subscribed organizations using this service.

## What does it looks like

****

**Figures 5.4 Asana Kanban Screenshot**

Asana also provide different view of the project (such as Board, Overview, List, Timeline, Calendar, Dashboard, etc), which the board view allows the development team to perform Kanban development, above screenshot is a sample from Asana, with one of the cards opened on the right panel, the card allow user to enter different attribute such as assigned member, task dependencies, priority, sub task, etc.

## Special Feature



**Figures 5.4 Asana Report Screenshot**

Asana has a built-in reporting page, allowing the management team to understand the current progress of the development team, this can partially lift the communication between project manager and different parties (such as management team and client).

Developers can check out their assigned task in their own personal “My Task” page, this can let users focus on their task at hand quickly, instead of finding their tasks among all the available tasks on the board.

Besides the reporting page, Asana also provides tools to automate the cards (using workflow tab), workflow can be considered as the lifecycle of each card, from user submitting request (maybe via google form, or github issue), going through different column (to do, doing , done), to archiving the card, this allow the cards to be processed systematically

## Cons

Asana does not have any obvious indication on the task-in-progress compared to other 2 tools, and requires the development team to ensure not too many tasks are at the processing stage.

**References**

1. <https://www.atlassian.com/agile/Kanban>
2. <https://www.atlassian.com/agile/Kanban/boards>
3. <https://www.amazon.com/Kanban-Successful-Evolutionary-Technology-Business/dp/0984521402>
4. [https://Kanbanize.com/Kanban-resources/getting-started/prioritizing-tasks-with-Kanban](https://kanbanize.com/Kanban-resources/getting-started/prioritizing-tasks-with-Kanban)
5. <https://global.toyota/en/company/vision-and-philosophy/production-system/>
6. <https://agilemanifesto.org/>
7. <https://www.sinnaps.com/en/project-management-blog/Kanban-board-example>
8. <https://dl-acm-org.ezproxy.cityu.edu.hk/doi/10.5555/41765.41801>
9. <https://pascal.computer.org/sev_display/search.action;jsessionid=9PBLPvqSSymDyg0BpCg4ldpBwE9dbo5QzYwSeoly.cslcpav04>
10. <https://ieeexplore-ieee-org.ezproxy.cityu.edu.hk/document/8100771>
11. <https://www.agilesherpas.com/blog/wip-limits-agile-marketing>
12. <https://netmind.net/en/principles-of-Kanban-its-much-more-than-just-post-its/>
13. <https://www.productplan.com/glossary/agile-principles/>
14. <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>
15. [https://Kanbantool.com/Kanban-software-development](https://kanbantool.com/Kanban-software-development)
16. [https://Kanbantool.com/use-cases/Kanban-in-software-development/time-tracking](https://kanbantool.com/use-cases/Kanban-in-software-development/time-tracking)
17. <https://www.extremeuncertainty.com/when-to-use-Kanban-vs-scrum/>
18. [https://www.atlassian.com/agile/Kanban/boards](https://www.atlassian.com/agile/kanban/boards)
19. [https://skilltec.ch/what-is-a-Kanban-board/](https://skilltec.ch/what-is-a-kanban-board/)
20. <https://www.planview.com/resources/articles/Kanban-101/>
21. [https://www.agilealliance.org/glossary/Kanban/#q=~](https://www.agilealliance.org/glossary/kanban/#q=~)(infinite~false~filters~(postType~(~'page~'post~'aa\_book~'aa\_event\_session~'aa\_experience\_report~'aa\_glossary~'aa\_research\_paper~'aa\_video)~tags~(~'Kanban))~searchTerm~'~sort~false~sortDirection~'asc~page~1)
22. [https://getnave.com/blog/Kanban-meetings/(Kanban-meeting)](https://getnave.com/blog/kanban-meetings/(kanban-meeting))
23. <https://global.toyota/en/company/vision-and-philosophy/production-system/>
24. [https://Kanbanize.com/Kanban-resources/case-studies/Kanban-for-qa-teams](https://kanbanize.com/kanban-resources/case-studies/kanban-for-qa-teams)
25. [https://www.tutorialspoint.com/agile\_testing/agile\_testing\_Kanban.htm](https://www.tutorialspoint.com/agile_testing/agile_testing_kanban.htm)
26. <https://kruschecompany.com/Kanban-method-agile-software-development/#How_does_the_Kanban_system_fit_within_the_Agile_approach_to_software_development>

-- END OF REPORT --

1. https://www.atlassian.com/agile/Kanban [↑](#footnote-ref-0)
2. https://www.atlassian.com/agile/Kanban/boards [↑](#footnote-ref-1)
3. https://www.atlassian.com/agile/Kanban/boards [↑](#footnote-ref-2)
4. https://www.atlassian.com/agile/Kanban/cards [↑](#footnote-ref-3)
5. https://Kanbanize.com/Kanban-resources/getting-started/prioritizing-tasks-with-Kanban [↑](#footnote-ref-4)
6. https://global.toyota/en/company/vision-and-philosophy/production-system/ [↑](#footnote-ref-5)
7. https://agilemanifesto.org/ [↑](#footnote-ref-6)
8. https://dl-acm-org.ezproxy.cityu.edu.hk/doi/10.5555/41765.41801 [↑](#footnote-ref-7)
9. https://pascal.computer.org/sev\_display/search.action;jsessionid=9PBLPvqSSymDyg0BpCg4ldpBwE9dbo5QzYwSeoly.cslcpav04 [↑](#footnote-ref-8)
10. https://ieeexplore-ieee-org.ezproxy.cityu.edu.hk/document/8100771 [↑](#footnote-ref-9)
11. <https://www.agilesherpas.com/blog/wip-limits-agile-marketing> [↑](#footnote-ref-10)
12. https://netmind.net/en/principles-of-Kanban-its-much-more-than-just-post-its/ [↑](#footnote-ref-11)
13. https://www.productplan.com/glossary/agile-principles/ [↑](#footnote-ref-12)
14. https://www.productplan.com/glossary/agile-principles/ [↑](#footnote-ref-13)
15. https://getnave.com/blog/Kanban-meetings/ [↑](#footnote-ref-14)
16. https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/ [↑](#footnote-ref-15)
17. https://netmind.net/en/principles-of-Kanban-its-much-more-than-just-post-its/ [↑](#footnote-ref-16)
18. https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/ [↑](#footnote-ref-17)
19. https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/ [↑](#footnote-ref-18)
20. https://www.crunchbase.com/organization/notion-so [↑](#footnote-ref-19)
21. https://www.notion.so/about [↑](#footnote-ref-20)
22. https://trello.com/about [↑](#footnote-ref-21)
23. https://trello.com/teams [↑](#footnote-ref-22)
24. https://asana.com/company [↑](#footnote-ref-23)