LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

**Experiment No.07**

**A.1 Aim:**

Introduction to JIRA Tool.

**A.2 Prerequisite:**

* Project Goals
* Target Audience
* Installed JIRA
* Size of the project
* Development team size

**A.3 Outcome:**

**After successful completion of this experiment students will be able to:**

1. Understanding the JIRA Tool
2. Understanding the Project Management tracking through JIRA tool
3. Understanding the Product backlogs
4. Understanding the KPIs of projects

**A.4 Theory:**

The main goal for this experiment to understand how to track, Control, Manage the project through JIRA tools

**Objectives of FPA:**

**A.5 Procedure/Algorithm:**

* Understand the Project Management tool JIRA
* Create the project under the JIRA

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

***(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Black board access available)***

| Roll No. | Name: |
| --- | --- |
| Class : | Batch : |
| Date of Experiment: | Date of Submission |
| Grade : |  |

**B.1 Answers of Task to be written by student:**

***(Paste your answers completed during the 2 hours of practical in the lab here)***

Kanban board in Manufacturing:

To successfully implement Kanban in manufacturing, you need to make sure you have the right environment. The further you are from the ‘ideal’ condition, the harder it will be to implement the system and the larger the safety margin you will need to add to the system to ensure there are no problems.

The ideal conditions for implementing Kanban in manufacturing are the following:

1. Have a constant predictable demand for product.
2. Have only a few product variations or use many common parts between variations and making the products distinctive at the later (finishing) stages.
3. Have an established [single] flow for all processes.
4. Use small(er) dedicated machines.
5. Able to make quick changeovers and/or use Single Minute Exchange of Die (SMED) techniques.
6. Have repeatable and reliable production processes and/or use [Total Productive Maintenance (TPM)](https://www.manufacturing.net/operations/blog/13058400/what-is-total-productive-maintenance), 5S, operator-driven quality improvements, and standardized operations.
7. Collaborate with reliable suppliers.

## Setting Up a Pull Production System:

Kanban is part of Just in Time (JIT) and Lean Manufacturing. As part of the pull system, it controls the what, when, and how much of the production. And it’s purpose is to ensure you are producing only as much as customers are asking for, and not a piece more. In order to do this, Kanban relies on JIT to create a pull system.

To do this, there are several Kanban rules that can help you implement Kanban and set up a pull production system:

* The later process gathers product from the previous process, and the earlier process
* The later process informs the previous process of what to manufacture
* The previous process produces only what the later process requires
* No products or materials can be moved or produced without Kanban authority
* Defects should be identified as close to the source as possible and are not passed to the later processes.

In reality, production is rarely as simple as manufacturing just one product at a time as it’s not economical nor fast enough even if rapid changeovers are done. Usually, production lines are used for making several products. And this can make it a bit more complicated to design the pull system appropriately.

## The Different Kanban Systems for Your Manufacturing System

The Kanban system you implement should fit your specific production needs and be designed accordingly. Some of the common approaches for applying Kanban in manufacturing that works for almost any business are:

### Kanban Cards and Kanban Bins

The Kanban cards approach uses simple cards which are attached to the batch of material as it moves through the production line. Very often, for each product, there are only two to three cards in the system. But if you need to handle larger batches, or if the product itself is big, there may be more cards associated with it. Typically, the cards contain information about the product – what it is, where to use it, and in what quantity it should be available.

The Kanban bins are used very similarly to the Kanban cards. But instead of using cards attached to the materials used in the process, the containers or bins that store them become the actual Kanban. When they are emptied, they are returned to the previous step as a signal to produce and be restocked. The bins are also labeled with similar information as the cards.

**CONWIP Systems**

The acronym stands for Continual Work In Progress. This approach to Kanban in manufacturing most closely resembles supermarket shelves restocking – the Kanban is the actual location of the material or part – the empty floor space or shelf. This system’s production schedule depends on demand. The CONWIP approach has higher lead times and helps manufacturers lower levels of inventory compared to the use of cards of bins.

But, the drastically reduced lead times are possible only if JIT delivery is already set up and if the system can achieve near to one-piece flow and variation is limited. With a ‘make-to-order’ approach to planning, CONWIP can also be successfully applied within areas with a large amount of variation.

**How to use kanban in software development**

We started with three columns six years back and look what we’ve got now. It’s a real Kanvolution (the evolution of a Kanban board, a term that we’ve created). Now let’s get the columns and swimlanes explained:

***Columns from left to right:***

**Backlog** – Ideas that we may or may not implement.

**Requested / Business Requirements** – Ideas that are going to be developed. The business need of the work item is listed inside the card.

**Requested / Ready for Tech Design** – Things that have been clarified and agreed to get to In Progress.

**In Progress / Tech Design / Tech Design** – The assignee of any card that is started writes in the technical design of what they’re about to start developing.

**In Progress / Tech Design/ Ready for Review** – The card is wating to be reviewed.

**In Progress / Tech Design/ Tech Design Review** – A fellow team member reviews the tech design and suggest changes (if necesarry).

**In Progress / Tech Design/ Ready for Coding** – After the tech design is finalized, the card remains here until the assignee can start coding.

**In Progress / Development / Coding** – Code is written.

**In Progress / Development / Ready for Code Review** – Cards are waiting for code review.

**In Progress / Development / Code Review** – The actual review stage where the code is checked.

**In Progress / Ready for Production** – Cards ready to be deployed on production.

**Production / To be Tested on Production** – Cards deployed on to production, pending a final verification by the QA team.

**Done / Possible to Return** – For special features requested by customers that may be returned for rework.

**Done** – Done pile for non-production related items like research, meetings, etc.

***Swimlanes from top to bottom:***

**Expedite** – Super urgent cards. Everyone helps to get the item expedited. Usually a critical customer defect, security-related issues, etc.

**Customer Issues** – Defects reported by customers. The customer is always first, remember?

**Bugs** – Internal bugs found by somebody on the team

**Technical Debt** – Things that we should have done

**Customer** / Business Features – New features to be introduced in Kanbanize

**Technical Features** – Usually DevOps tasks, deployment-related activities, database schema changes, etc.

It’s important to note that all swimlanes represent priority. The top swimlane is the highest priority and the bottom one is the lowest priority. One of the main principles of Lean software development is **Build-in-quality**. In other words, **you shouldn’t allow defects to occur**. This is why you need to fix any issues that you find immediately. That’s only achievable with great focus and dedication (and a great support team, of course). This is why it is worth trying Kanban in software development. Because it shows you everything and helps your team stay focused.

Let’s give a bit more details about the process involved in each of the swimlanes, as it’s important to note the actual behavioral differences related to the separate groups.

**Expedite** is pretty much the most critical area on the board. In fact, it is treated with such priority, that everyone stops doing what they’re doing and helps to get the issue expedited. A sample expedite item would be a production-down situation, significant performance degradation, a hacker’s attack, key customer escalation, board-level escalation, etc.

**Customer Issues** are the real health indicator of your product or service. If you get huge amounts of issues piling up in the “Requested” area of your Kanban board, you have a very urgent problem to solve. Never ever allow that to happen, otherwise, you’ll soon be in serious trouble.

Customer issues are the single most important thing on your Kanban board no matter how minor they are and how stupid they may seem.

**Bugs (internal issues)** are not as critical as customer issues, but that’s a trap that most of the teams fall into. There’s a widespread saying “If the customer hasn’t seen it, it doesn’t exist”, which is partially true, but not the preferred line of thought as far as we’re concerned.

First, when you stop piling up issues they don’t grow old and you control the size of the backlog for open issues.

Second, at some point in time, the old issues will be too old to fix, which would give you the holy right to simply discard them, which combined with the fact that you’ve continuously prevented new issues from aging, would result in emptying your bugs backlog. Can you imagine zero issues on your backlog? Can you?

**Technical Debt (TD)** is a somewhat abstract term. This is a flaw in the product design/architecture that would eventually [lead to longer cycle times](https://kanbanize.com/kanban-resources/kanban-software/kanban-lead-cycle-time/) and high complexity in development and support. In fact, some product architectures are so bad that companies simply decide to ditch them and re-create the product from scratch.

**Business Features is the fourth swimlane, which makes it less important than the other three above, but it’s definitely very important.** This is where value gets generated, and since we’re usually paid to generate value, it’s crucial that we flow work through this swimlane in a fast and predictable manner. In fact, the swimlanes above are a sort of investment that we make, for the sake of being able to consistently produce value without being swarmed with customer complaints, escalations and firefighting activities in the long run.

**Technical Features** are lower on the list of priorities, but again, it doesn’t mean they’re unimportant. As a matter of fact, technical features may often make it to the expedited swimlane, if we have an automation piece that blocks the release.

The main purpose of this swimlane is to visually separate technical features from business features and to allow better tracking of the given work type.

Theoretically, you could merge this swimlane with the business features one, but we prefer to have it on its own because the team members that operate in it are usually different from the ones that work on the business features.

Basically, this is how Kanban works for software development teams.

Now let’s check how to limit your work in progress.

**Kanban in Toyota:**

Consider a pull system in a Toyota assembly plant. Orders accumulate from car dealerships. Production control creates a leveled schedule. For example, they make a white Camry, followed by a green Camry, followed by a Red Avalon, and so on. Each of these cars has a whole set of options associated with it. That schedule is sent to the body shop, where stamped steel panels (from a “supermarket” of pre-stamped panels) are welded together into a body. Stamping the panels is a much faster operation than the takt time in assembly plants (e.g., one second per stamped panel versus 60-second takt times in a plant is typical), so putting them into a one-piece flow is not practical. They would be productive one out of every 60 seconds. So a pull system is used. At a certain trigger point when a certain number of steel panels have been used by the body shop, a kanban goes back to a stamping press, ordering it to make another batch to replenish the store.

Similarly, when assembly line workers begin to use parts from bins (hinges, door handles, windshield wipers), they take out a kanban card and put it in a mailbox. A material handler will come on a timed route and pick it up and go back to a store to replenish what is used on the assembly line. Another material handler will replenish the store based on parts from a supermarket of supplier parts. This will trigger an order back to parts suppliers. And so on. Illustrates a system like this, where parts in the assembly plant are replenished from a supplier. The process starts at the assembly factory, then “withdrawal kanban” and empty containers are sent by truck back to the supplier to be refilled. The supplier keeps a small store of finished parts in a “parts store,” which are used to refill the kanban and their empty containers. When parts are withdrawn from the parts store shelves, they must be replenished by sending a kanban and an empty container back to the production cell where new parts are built and then sent to refill the “parts store” shelves. Information, orders for parts in the form of kanban, flows backward from the customer (the assembly plant). Materials, in this case parts, are sent forward to the customer.

**B.2 Observations and learning:**

***(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)***

**B.3 Conclusion:**

*(****Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)***

<https://www.ineak.com/toyotas-kanban-systempull-where-you-must/>