

# **SAGAR INSTITUTE OF SCIENCE & TECHNOLOGY, BHOPAL**



## **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

### **SUBJECT-AGILE SOFTWARE DEVELOPMENT(ASD)**

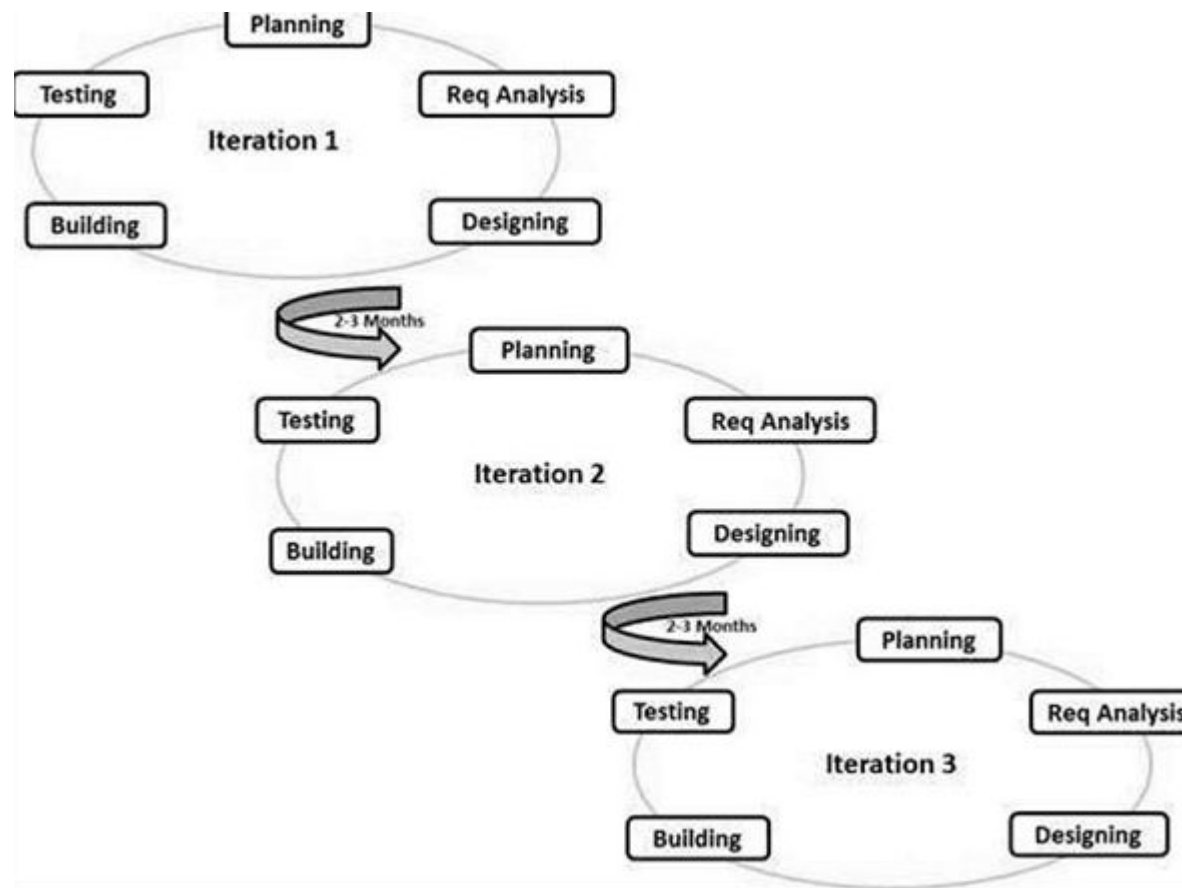
### **SUBJECT CODE-CS-703(C)**

# UNIT-I

# AGILE CONECEPT

- ◉ Agile is the ability to create and respond to change.
- ◉ It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment.
- ◉ represented the adaptiveness and response to change which was so important to their approach.
- ◉ Agile based on the incremental & Iterative approach.

# AGILE



# AGILE PROCESS

- ◉ **Individuals and interactions** – In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
- ◉ **Working software** – Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
- ◉ **Customer collaboration** – As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
- ◉ **Responding to change** – Agile Development is focused on quick responses to change and continuous development.

# AGILE PRINCIPLES

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.

# AGILE PRINCIPLES

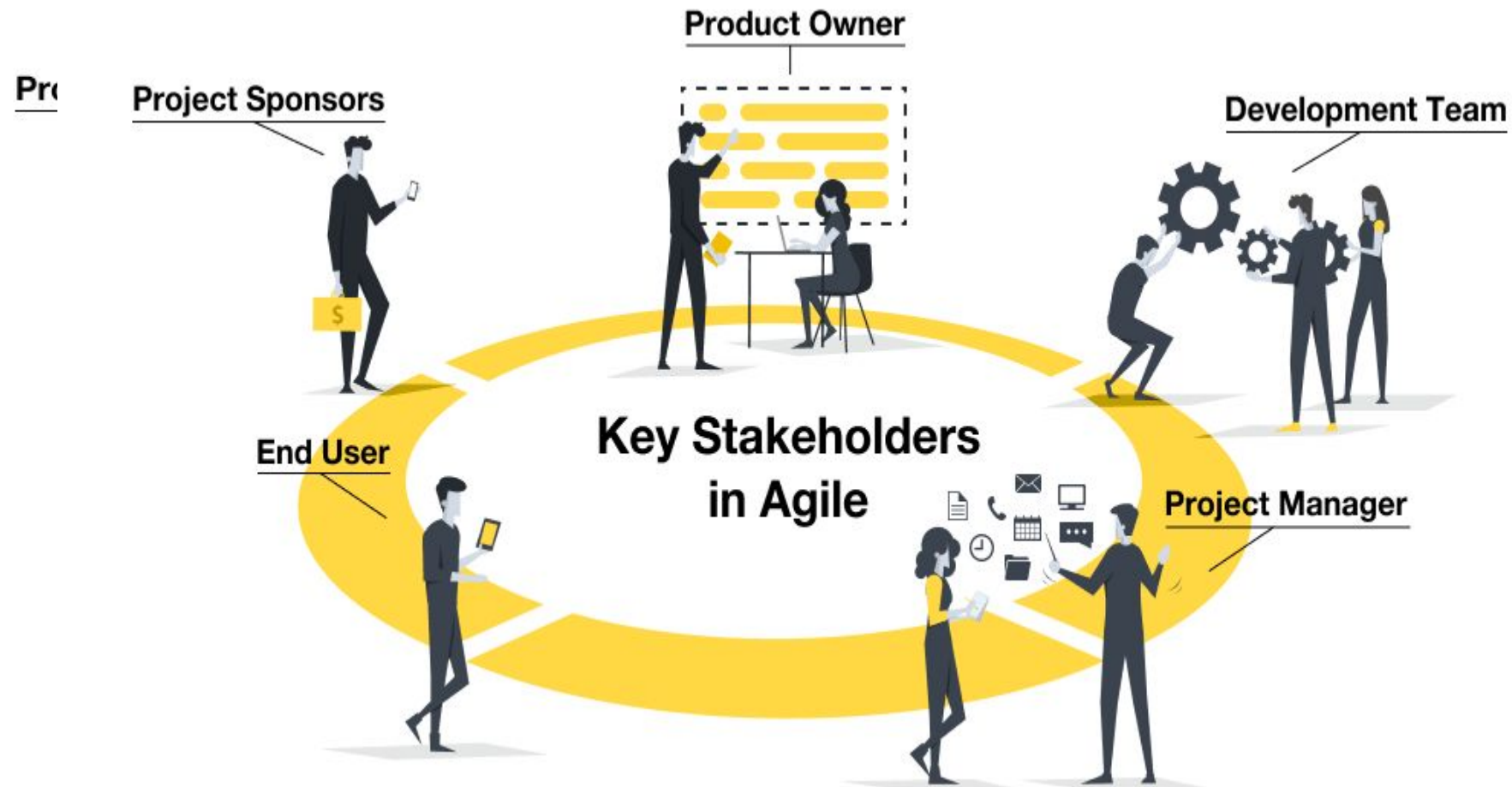
- 5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- 6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- 7. Working software is the primary measure of progress.
- 8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- 9. Continuous attention to technical excellence and good design enhances agility.
- 10. Simplicity-the art of maximizing the amount of work not done-is essential.

# STAKEHOLDER

- ◉ Stakeholder(s) are people and organization units who frequently interface with the product owner, and team to provide them with inputs and facilitate creation of the project's products and services.
- ◉ influencing the project throughout the project's development. Typically, stakeholder includes customers, users, and sponsors.



# STAKEHOLDER IN AGILE



- ◉ In Agile development stakeholders can be represented by a wide range of people interested in project results:
- ◉ People funding the project;
- ◉ Business Managers and Business Architects;
- ◉ Data Architects and Database Administrators;
- ◉ Portfolio and Project Managers;
- ◉ Direct and indirect Users;
- ◉ Account and Sales Managers;
- ◉ Developers' team including Engineers, Designers and PM/BA, etc.

# CHALLENGE

- ◉ Processes / procedures an obstacle to productivity.
- ◉ Communicating effectively with stakeholders and business owners
- ◉ Adapting to changes in business requirements while minimizing wasted effort
- ◉ Unrealistic initial expectations
- ◉ Unclear declaration
- ◉ of focus
- ◉ the mismatch of the culture of the organization and its people

# SOLUTIONS

- ◉ Cooperate to overcome the immediate challenge, then work together to form new processes that meet everyone's needs
- ◉ Including stakeholders in each phase of a project.
- ◉ Working iteratively

# Scrum Methodology

# SCRUM

- ◉ Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one.
- ◉ A “process framework” is a particular set of practices that must be followed in order for a process to be consistent with the framework. (For example, the Scrum process framework requires the use of development cycles called Sprints, the XP framework requires pair programming, and so forth.)
- ◉ “Lightweight” means that the overhead of the process is kept as small as possible, to maximize the amount of productive time available for getting useful work done.

# SCRUM

- ◉ Scrum is a framework that allows for more effective collaborations among teams working on complex projects.
- ◉ Agile and scrum are two similar project management systems with a few key differences.
- ◉ Agile is more flexible and promotes leadership teams, while scrum is more rigid and promotes cross-functional teams.





# BENEFIT FROM SCRUM

- ◉ **Complicated projects:** Scrum methodology is ideal for projects that require teams to complete a backlog.
- ◉ **Companies that value results:** Scrum is also beneficial to companies that value results over the documented progress of the process.
- ◉ **Companies that cater to customers:** Scrum can help companies that develop products in accordance with customer preferences and specifications.

# DIFFERENT ROLES IN AGILE SCRUM METHODOLOGY

- ◉ Scrum master
- ◉ Product owner
- ◉ Scrum team
- ◉ Product backlog
- ◉ Sprint review meeting
- ◉ Sprint retrospective

# SCRUM METHODOLOGY

- ◎ **Scrum team:** A typical scrum team has between five and nine people, but Scrum projects can easily scale into the hundreds. However, Scrum can easily be used by one-person teams and often is. This team does not include any of the traditional software engineering roles such as programmer, designer, tester or architect. Everyone on the project works together to complete the set of work they have collectively committed to complete within a sprint. Scrum teams develop a deep form of camaraderie and a feeling that “we’re all in this together.”
- ◎ **Product owner:** The product owner is the project’s key stakeholder and represents users, customers and others in the process. The product owner is often someone from product management or marketing, a key stakeholder or a key user.

# SCRUM METHODOLOGY

- ◎ **Scrum Master:** The Scrum Master is responsible for making sure the team is as productive as possible. The Scrum Master does this by helping the team use the Scrum process, by removing impediments to progress, by protecting the team from outside, and so on.
- ◎ **Product backlog:** The product backlog is a prioritized features list containing every desired feature or change to the product. Note: The term “backlog” can get confusing because it’s used for two different things. To clarify, the product backlog is a list of desired features for the product. The sprint backlog is a list of tasks to be completed in a sprint.

# SCRUM METHODOLOGY

**Sprint planning meeting:** At the start of each sprint, a sprint planning meeting is held, during which the product owner presents the top items on the product backlog to the team. The Scrum team selects the work they can complete during the coming sprint. That work is then moved from the product backlog to a sprint backlog, which is the list of tasks needed to complete the product backlog items the team has committed to complete in the sprint.

**Daily Scrum:** Each day during the sprint, a brief meeting called the daily scrum is conducted. This meeting helps set the context for each day's work and helps the team stay on track. All team members are required to attend the daily scrum.

# SCRUM METHODOLOGY

- ◎ **Sprint review meeting:** At the end of each sprint, the team demonstrates the completed functionality at a sprint review meeting, during which, the team shows what they accomplished during the sprint. Typically, this takes the form of a demonstration of the new features, but in an informal way; for example, PowerPoint slides are not allowed. The meeting must not become a task in itself nor a distraction from the process.
- ◎ **Sprint retrospective:** Also at the end of each sprint, the team conducts a sprint retrospective, which is a meeting during which the team (including its ScrumMaster and product owner) reflect on how well Scrum is working for them and what changes they may wish to make for it to work even better.

## DIFFERENCES BETWEEN SCRUM AND AGILE

Although scrum and agile are similar, here are some of the key differences:

- ◉ Scrum values rigidity, whereas agile is more flexible.
- ◉ Agile leaders play a vital role, while scrum promotes a cross-functional team that is self-functioning.
- ◉ Agile involves face-to-face interactions between cross-functional team members, while scrum involves daily stand-up meetings.
- ◉ Agile is meant to be kept simple, while scrum can be innovative and experimental
- ◉

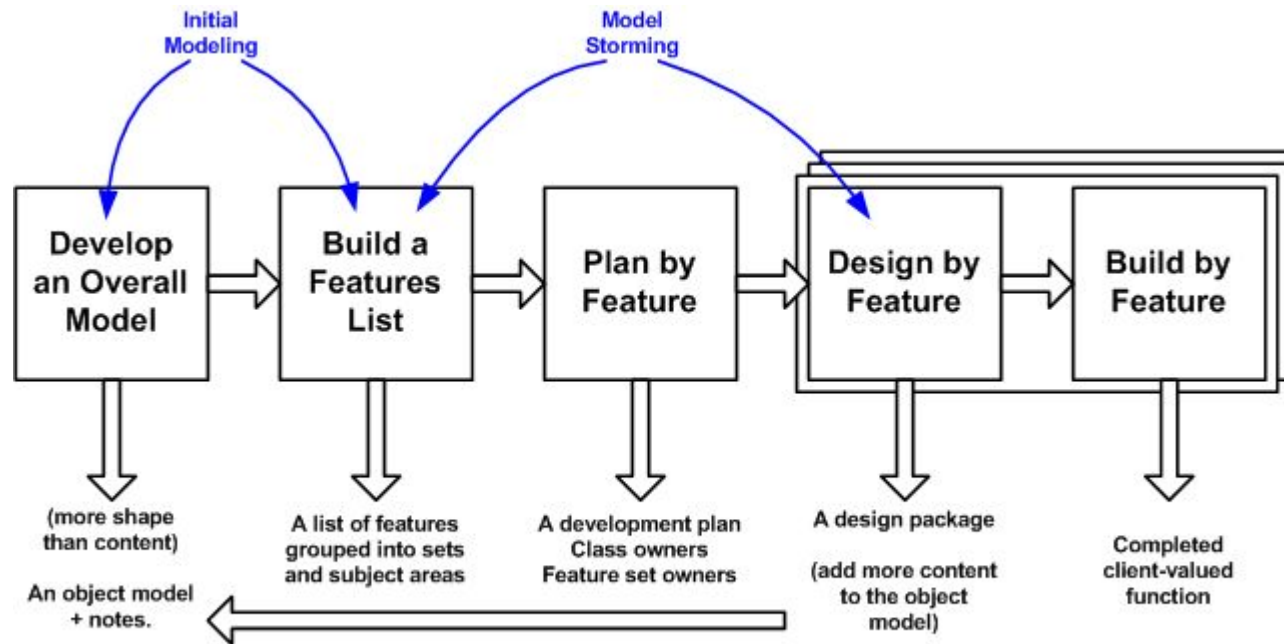
# Feature Driven Development



# FEATURE-DRIVEN DEVELOPMENT (FDD)

- ◉ Feature-Driven Development (FDD) is customer-centric, iterative, and incremental, with the goal of delivering tangible software results often and efficiently. FDD in Agile encourages status reporting at all levels, which helps to track progress and results.
- ◉ FDD allows teams to update the project regularly and identify errors quickly. Plus, clients can be provided with information and substantial results at any time.

# FEATURE-DRIVEN DEVELOPMENT (FDD)



Copyright 2002-2005 Scott W. Ambler  
Original Copyright S. R. Palmer & J.M. Felsing

# HOW DOES FDD WORK

- ◉ Develop overall model
- ◉ Build feature list
- ◉ Plan by feature
- ◉ Design by feature
- ◉ Build by feature

# STAGES OF FEATURE-DRIVEN DEVELOPMENT

- ◉ Gather Data
- ◉ Develop an overall model
- ◉ Build a features list
- ◉ Design by Feature
- ◉ Build by Feature

# FDD'S STRENGTHS INCLUDE

- ◉ Simple five-step process allows for more rapid development
- ◉ Allows larger teams to move products forward with continuous success
- ◉ Leverages pre-defined development standards, so teams are able to move quickly

# FDD'S WEAKNESSES INCLUDE

- ◉ Does not work efficiently for smaller projects
- ◉ Less written documentation, which can lead to confusion
- ◉ Highly dependent on lead developers or programmers

# BEST PRACTICES FOR FEATURE-DRIVEN DEVELOPMENT

- ◉ Identifying the domain object model, or the scope of the problem that needs to be solved, to help with the framework for feature development.
- ◉ Breaking down complex features into smaller functions and subsets.
- ◉ Assigning features to a single owner to ensure consistency and code integrity.
- ◉ Building dynamic and diverse feature teams to collect multiple design options.
- ◉ Performing routine code inspections of each feature before implementation into the main build.
- ◉ Enforcing project visibility with frequent, accurate progress reports during all steps.

# FEATURE DRIVEN DEVELOPMENT VS. SCRUM

- ◉ In Feature Driven Development, the actual user is viewed as the end-user, while in Scrum, the product owner is seen as the end-user. FDD additionally takes a shorter sprint length than in Scrum. FDD is class ownership, while Scrum is shared code ownership.
- ◉ FDD specifies the engineering practices, i.e. design/code, inspections, and tests, whereas Scrum doesn't specify any particular engineering practices, although parts of XP frequently use. FDD is domain-driven, while Scrum focuses on producing vertical slices of functionality accepted by the product owner.
- ◉ Additionally, in FDD feature teams have recognized roles (project manager, chief architect, development manager, chief programmer, class ownership, and domain expert), whereas Scrum is a self-organizing team.



# SIMILARITY BETWEEN FDD & SCRUM

- ◉ They both emphasize producing quality components and help track progress at different granularities
- ◉ Feedbacking
- ◉ Applicable only for complex projects.

# Crystal & Kanban

# CRYSTAL METHODS

- ◉ Crystal method is a agile framework that is considered as a lightweight or agile methodologies
- ◉ focuses on individuals and the interactions.
- ◉ It is mainly for short-term projects by a team of developers working out of a single workspace.

# HISTORY OF CRYSTAL METHOD

- ◉ Crystal method was developed by an American scientist named Alistair Cockburn who worked in IBM. He decided not to focus on step-by-step developmental strategies, but to develop team collaboration and communication. Some of the traits of Cockburn's Crystal method were:
- ◉ Human-powered i.e. the project should be flexible and people involved in preferred work.
- ◉ Adaptive i.e. approaches doesn't any fixed tools but can be but can be changed anytime to meet team's specific needs.
- ◉ Ultra-light i.e. this methodology doesn't require much documentation.

# PROPERTIES OF CRYSTAL AGILE FRAMEWORK

- ◉ Frequent Delivery
- ◉ Reflective Improvement
- ◉ Osmotic Communication
- ◉ Personal Safety
- ◉ Focus
- ◉ Easy access to expert users
- ◉ Technical tooling

# PROPERTIES OF CRYSTAL AGILE FRAMEWORK

- ◉ **Frequent Delivery-**

It allows you regularly deliver the products, test code to real users. Without this, you might build a product that nobody needs.

- ◉ **Reflective Improvement-**

No matter how good you have done or how bad you have done. Since there are always areas where the product can be improved, so the teams can implement to improve their future practices.

# PROPERTIES OF CRYSTAL AGILE FRAMEWORK

- ◉ **Osmotic Communication-**

Alistair stated that having the teams in a same physical phase is very much important as it allows information to flow in between members of a team as in osmosis.

- ◉ **Personal Safety-**

There are no bad suggestions in a crystal team, team members should feel safe to discuss ideas openly without any fear.

# PROPERTIES OF CRYSTAL AGILE FRAMEWORK

- ◉ **Focus-**  
Each member of team knows exactly what to do, which enables them to focus their attention. This boosts team interaction and work towards the same goal.
- ◉ **Easy access to expert users-**  
It enhances the team communication with users and get regular feedback from real users.
- ◉ **Technical tooling-**  
It contains very specific technical tools which to be used by software development team during testing, management and configuration. These tools make it enable for the team to identify any error within less time.



# HOW DOES CRYSTAL FUNCTION

Crystal family consists of many variants like Crystal Clear, Crystal Yellow, Crystal Red, Crystal Sapphire, Crystal Orange Crystal Web, Crystal Diamond.

## **Crystal Clear-**

The team consists of only 1-6 members that is suitable for short-term projects where members work out in single workspace.

## ◉ **Crystal Yellow-**

It has a small team size of 7-20 members, where feedback is taken from Real Users. This variant involves automated testing which resolves bugs faster and reduces use of too much documentation.

# HOW DOES CRYSTAL FUNCTION

- ◎ **Crystal Orange-**

It has a team size of 21-40 members, where team is split according to their functional skills. Here the project generally lasts for 1-2 years and the release is required every 3 to 4 months.

- ◎ **Crystal Orange Web-**

It has also the team size of 21-40 members where the projects that have a continually evolving code base that is being used by the public. It is also similar to Crystal Orange but here they do not deal with single project but series of initiatives that required programming.

# HOW DOES CRYSTAL FUNCTION

- ◉ **Crystal Red-**

The software development is led by 40-80 members where the teams can be formed and divided according to requirements.

- ◉ **Crystal Maroon-**

It involves large sized projects where team size is of 80-200 members where methods are different and as per the requirement of the software.

- ◉ **Crystal Diamond & Sapphire-**

This variant is used in large projects where there is a potential risk to human life.

# CRYSTAL TEAM SIZE



## **Benefits of using the Crystal Agile Framework :**

- ◉ Facilitate and enhance team communication and accountability.
- ◉ The adaptive approach lets the team respond well to the demanding requirements.
- ◉ Allows team to work with one they see the most effective.
- ◉ Teams talk directly with each other, that reduce management overhead.

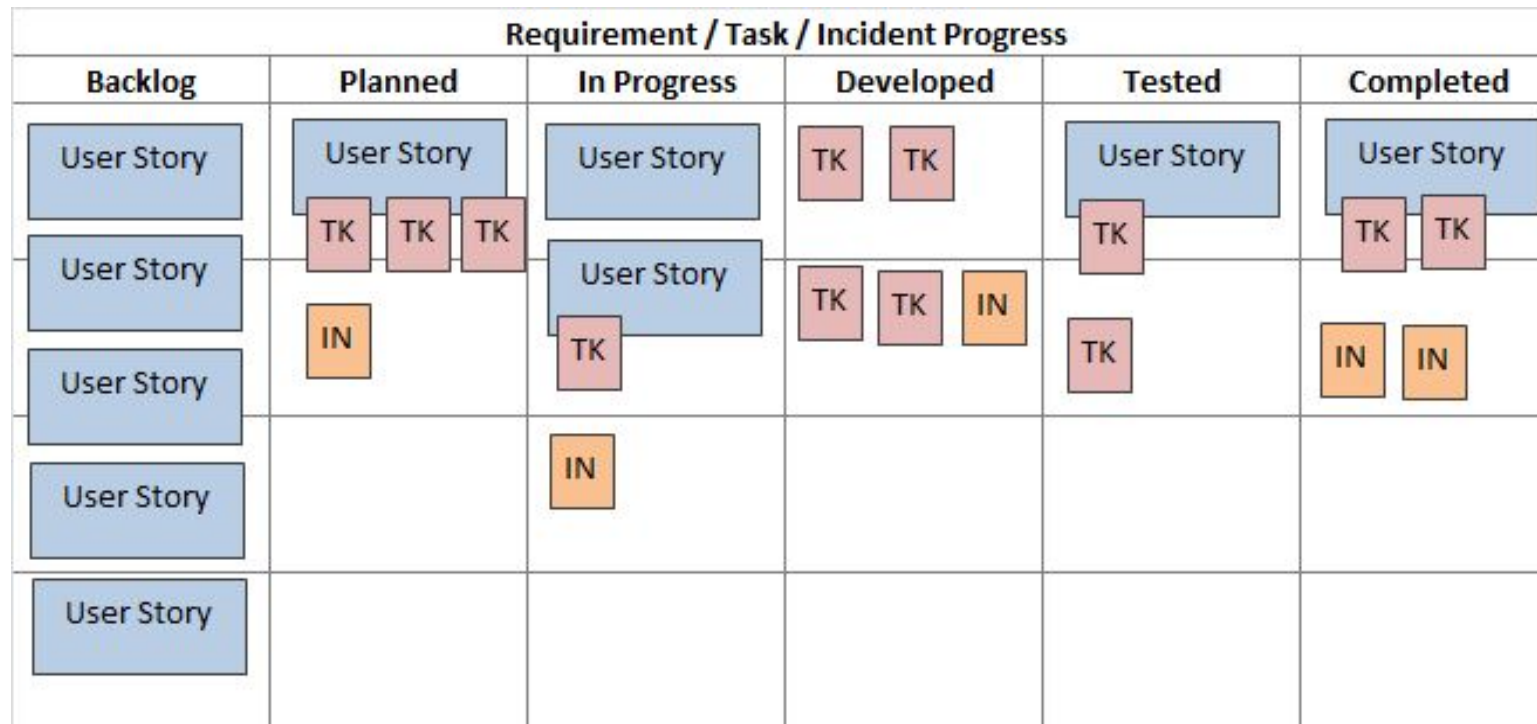
## **Drawbacks of using the Crystal Agile Framework :**

- ◉ Lack of pre-defined plans may lead to confusion and loss of focus.
- ◉ Lack of structure may slow down inexperienced teams.
- ◉ Not clear on how a remote team can share knowledge informally.

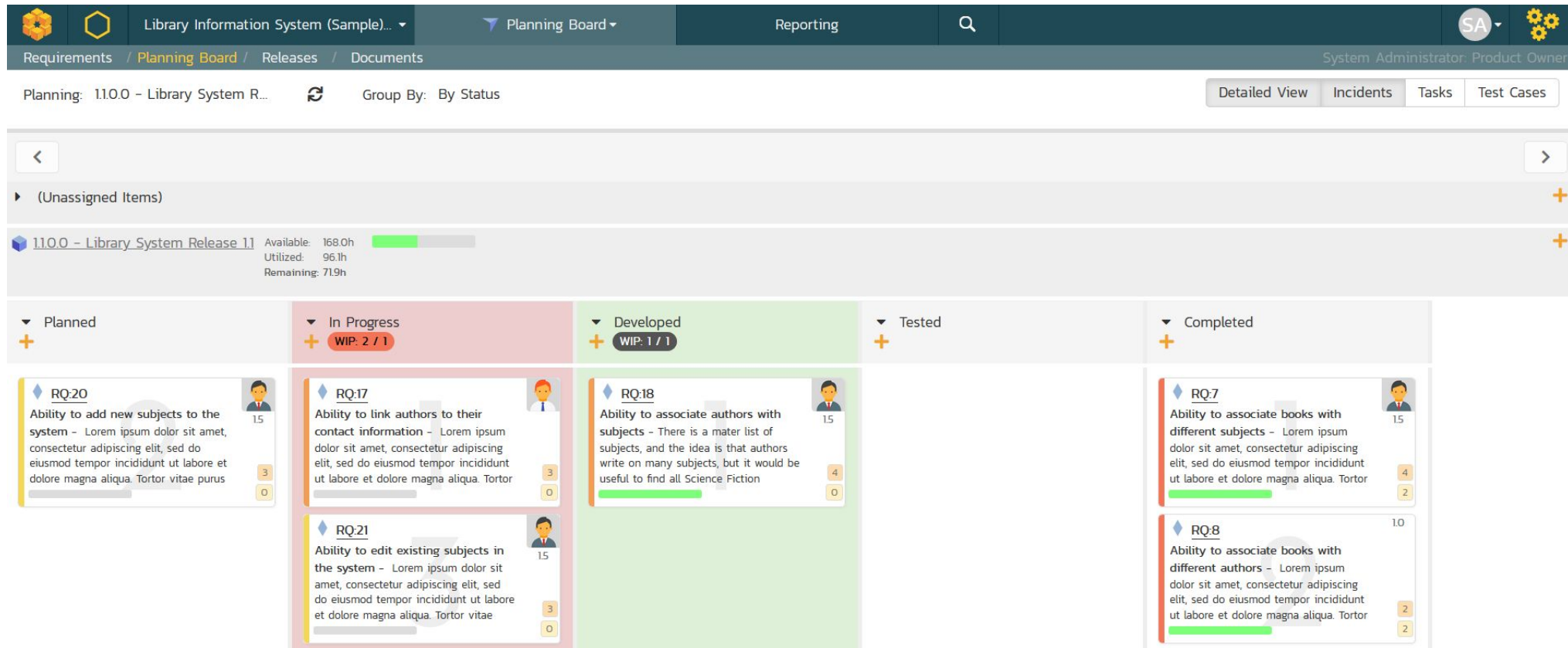
# KANBAN IN AGILE

- ◉ Kanban is a popular framework used to implement agile and DevOps software development.
- ◉ It requires real-time communication of capacity and full transparency of work.
- ◉ Work items are represented visually on a kanban board, allowing team members to see the state of every piece of work at any time.

# KANBAN IN SOFTWARE DEVELOPMENT

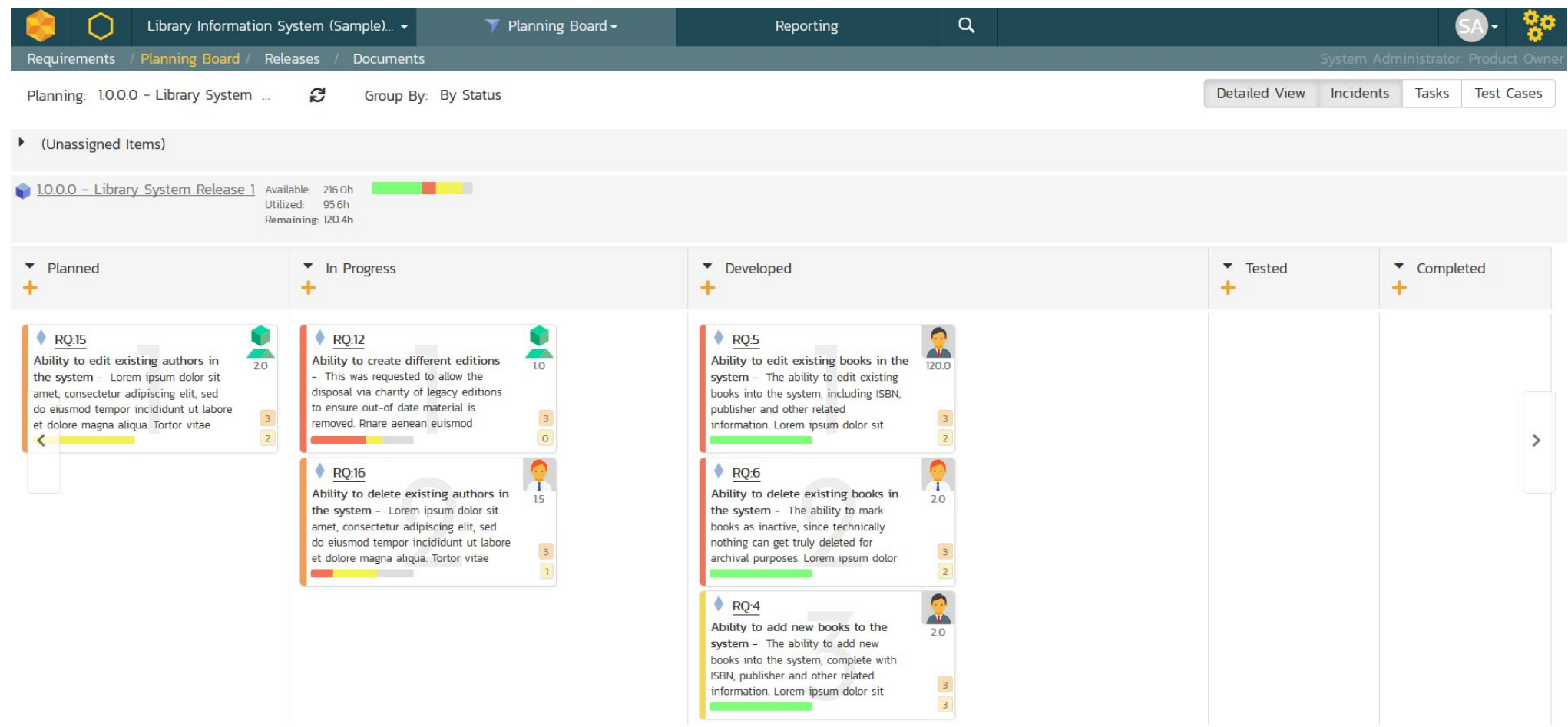


# KANBAN: A SIMPLE WAY TO MANAGE WORK









# KANBAN BOARDS






# KANBAN BOARDS

- ◉ SpiraTeam provides a Kanban view of the project, where you can see all of the requirements planned for each release organized according to their position in the lifecycle. This view lets you see the flow of the requirements and identify then rectify any bottlenecks:
- ◉ SpiraTeam has dedicated Kanban boards for tasks and incidents / defects, so that it makes it easy to run both development and maintenance projects using the same system:












































# KANBAN BOARDS

 Library Information System (Sample) ... Incidents Reporting  SA 

Incidents / Tasks / Resources / Source Code System Administrator Product Owner

Planning: (All Releases)  Group By: By Status Detailed View  

▶ (Unassigned Items)

New +	Assigned +	Open +	Reopen +	Closed +	Duplicate +
<div> <b>IN:1</b> Cannot log into the application</div>	<div> <b>IN:7</b> Cannot add a new book to the system  0.3h</div>	<div> <b>IN:4</b> Database not backing up correctly</div>		<div> <b>IN:15</b> Session handling  0.5h</div>	<div> <b>IN:17</b> Cannot add a new book to the system </div>
<div> <b>IN:2</b> Not able to add new author</div>	<div> <b>IN:21</b> Ability to associate multiple authors  0.4h</div>	<div> <b>IN:18</b> Ability to integrate with other systems</div>		<div> <b>IN:35</b> Section 508 compliance  0.9h</div>	<div> <b>IN:37</b> System may require process changes</div>
<div> <b>IN:3</b> Clicking on link throws fatal error </div>	<div> <b>IN:40</b> Test Training Item  0.5h</div>	<div> <b>IN:20</b> Ability to remember passwords</div>		<div> <b>IN:55</b> Test Change Request  0.9h</div>	<div> <b>IN:44</b> Test Training Item  &gt;</div>
	<div> <b>IN:46</b> Test System Limitation  0.8h</div>	<div> <b>IN:30</b> Cannot install system on Windows 10</div>		<div> <b>IN:36</b> How do we regression test  0.8h</div>	<div> <b>IN:29</b> Ability to associate multiple authors</div>
	<div> <b>IN:8</b> Editing the date on a book is clunky  0.4h</div>	<div> <b>IN:5</b> Cannot install system on Oracle 9i</div>		<div> <b>IN:56</b> Test Change Request  1.3h</div>	<div> <b>IN:50</b> Test System Limitation </div>
	<div> <b>IN:22</b> Integration of security with NTLM/PKI  0.4h</div>	<div> <b>IN:45</b> Test System Limitation 0.5h</div>		<div> <b>IN:13</b> The tables get cutoff on low-res modes  0.3h</div>	<div> <b>IN:57</b> Test Change Request</div>




# WORK IN PROGRESS (WIP)

- ◉ Work In Progress (WIP) limits set the maximum number of requirements that the product team can efficiently manage at each stage of their Kanban process. Using WIP limits can be a useful way for teams to manage their work, allowing them to get through their work faster. This is done by focusing only tasks that can be done now (in other words, the work that can in-progress at any one time).

# WORK IN PROGRESS (WIP)

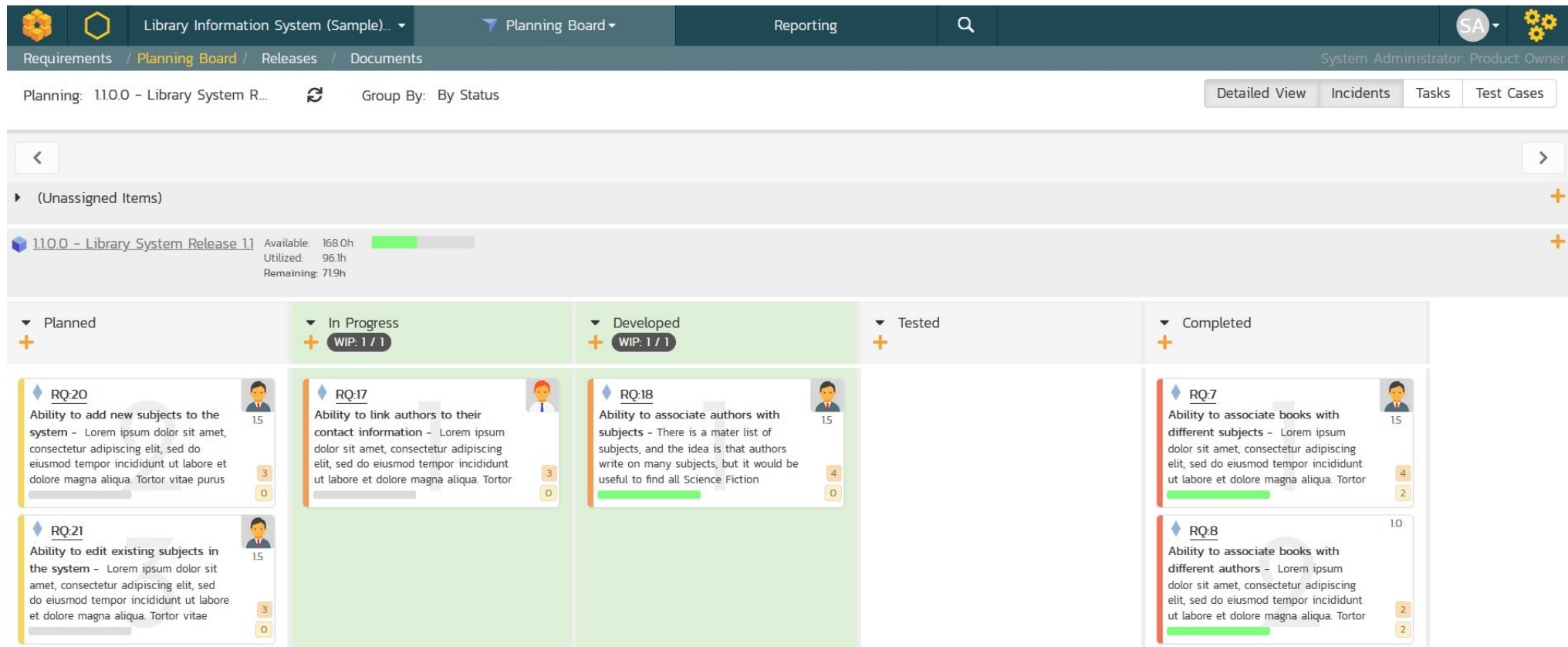
## Kanban Work in Progress (WIP) Limits

Please enter the % Work in Progress (WIP) limits for each requirement status (blank means no limit), as well as the multiplier that specifies how many requirements should be active in the entire release/iteration based on the # resources allocated to the specific release/sprint:

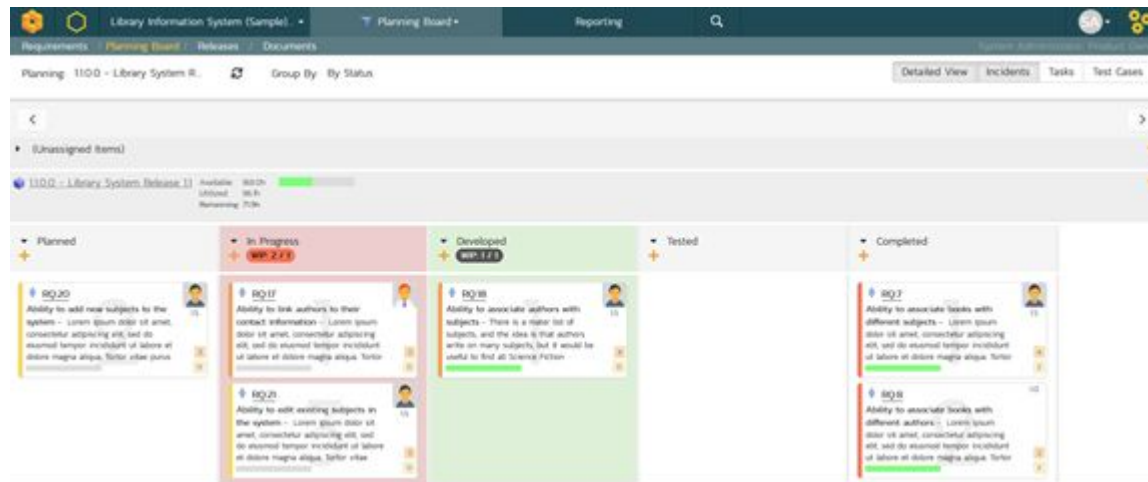
Status	 Release WIP	 Sprint WIP
Planned:	<input type="text"/> %	<input type="text"/> %
In Progress:	<input type="text" value="100.0"/> %	<input type="text"/> %
Developed:	<input type="text" value="200.0"/> %	<input type="text"/> %
Tested:	<input type="text"/> %	<input type="text"/> %
Completed:	<input type="text"/> %	<input type="text"/> %
WIP Multiplier:	<input type="text" value="1.0"/>  	<input type="text" value="1.0"/>  

**i** For example, if you have a sprint with 5 people, you can set the WIP multiplier to be 2x the number of people. That will allow 10 requirements to be active in the sprint. You can then specify that Developed and Tested both allow 50% of the WIP items. That means that both Developed and Tested will allow upto 5 items in the Kanban board.

# WORK IN PROGRESS (WIP)



# WORK IN PROGRESS (WIP)



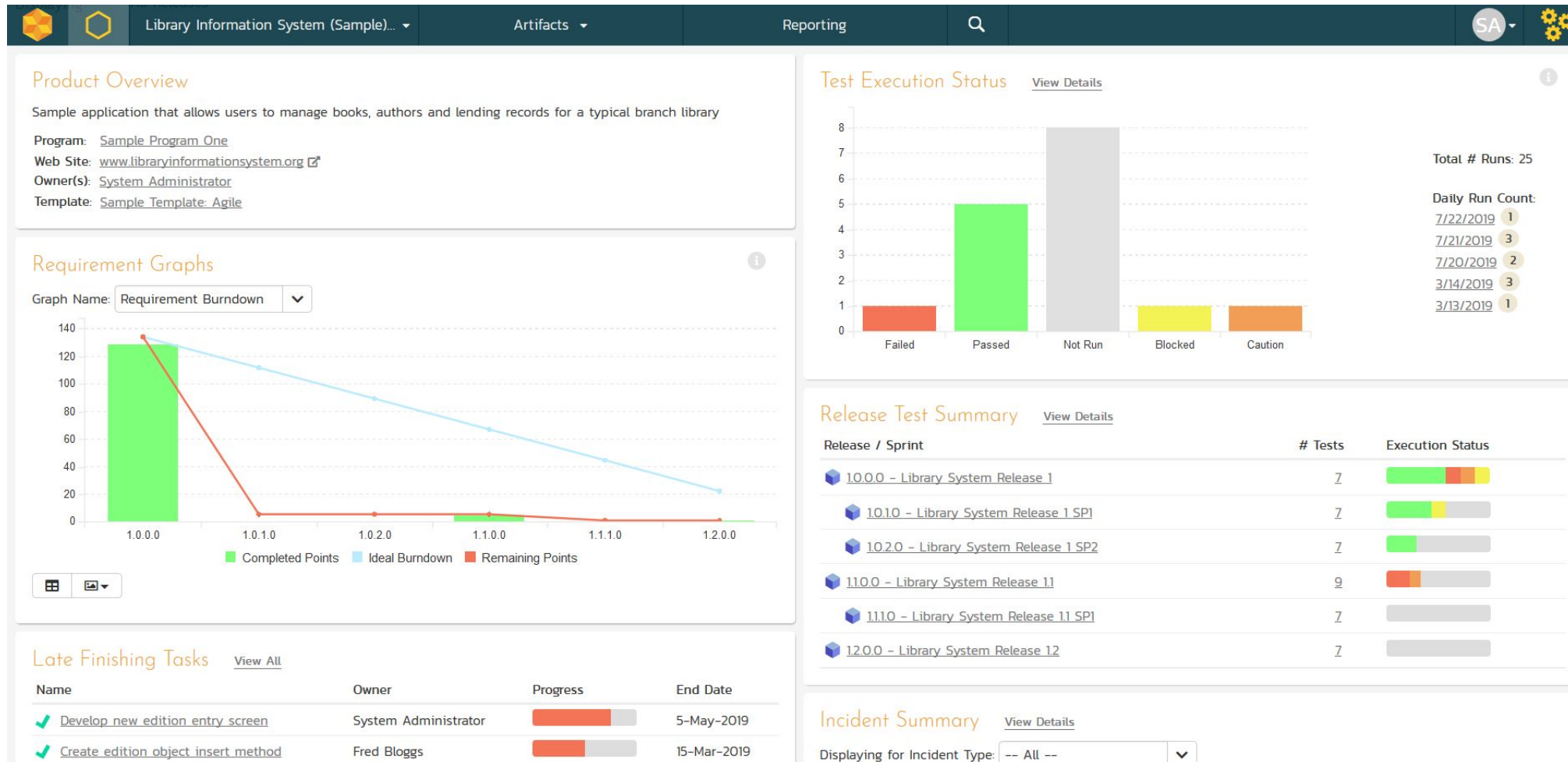


# REPORTING DASHBOARDS

- ◉ SpiraTeam® provides reporting dashboards of key project quality and progress indicators - requirements test coverage, task progress, project velocity, as well as top risk and issues - in one consolidated view that is tailor-made for Kanban projects as well as supporting your legacy/hybrid waterfall projects.



# REPORTING DASHBOARDS



# REASONS

- ◉ The top reasons that our customers choose SpiraTeam® over other Kanban project management solutions are:
- ◉ It is a **complete Kanban project management** solution that includes requirements management, release planning, task, task estimation and defect tracking all fully integrated from day one.
- ◉ **Highly intuitive** web application that provides a **complete picture** of a project's status and health yet requires only a web-browser.
- ◉ Available as both a **SaaS/hosted** service and an **on-premise** deployment, SpiraTeam is both affordable and easy to deploy for your Kanban projects.
- ◉ Ability to **leverage your existing technology investments**. SpiraTeam integrates with many third-party defect-management systems and software configuration management (SCM) systems as well as your existing build server and unit test frameworks.
- ◉ In addition, we provide superb technical support that ensures that enquiries and questions are dealt with in a timely and professional manner.

# LEAN SOFTWARE DEVELOPMENT

# LEAN SOFTWARE DEVELOPMENT

- Lean software development is a set of principles that can be applied to software development to decrease programming effort, budgeting, and defect rates by one third.

# LEAN SOFTWARE DEVELOPMENT



# SOFTWARE DEVELOPMENT PRINCIPLES

- ◉ Waste Elimination
- ◉ Amplifying Learning
- ◉ Late Decision Making
- ◉ Fast Delivery
- ◉ Team Empowerment
- ◉ Built-in Integrity
- ◉ View Applications as a Whole

# ELIMINATE WASTE

- ◉ One of the key elements of practicing Lean is to eliminate anything that does not add value to the customer. There are seven wastes (or muda) defined in the Toyota school of Lean manufacturing. They are:
- ◉ Over-production: developing an item before it is required
- ◉ Unnecessary transportation: Moving inventory from place to place, which puts it at risk for damage without adding any value
- ◉ Inventory: Holding inventory adds cost without adding any value to the customer; excess inventory takes up valuable space, increases lead times, and delays innovation
- ◉ Motion: Literally refers to unnecessary movement of workers on the shop floor
- ◉ Defects: Quality issues result in rework or scrap and can add tremendous additional costs to organizations who don't habitually find ways to eliminate sources of defects
- ◉ Over-processing: Using advanced, expensive tools to do what could be done with simpler tools

- ◉ Waiting: When inventory waits between value-adding steps
- ◉ Tom and Mary Poppendieck translated those wastes to software development. Each of these wastes should be systematically eliminated in order to maximize customer value:
- ◉ Unnecessary code or functionality: Delays time to customer, slows down feedback loops
- ◉ Starting more than can be completed: Adds unnecessary complexity to the system, results in context-switching, handoff delays, and other impediments to flow
- ◉ Delay in the software development process: Delays time to customer, slows down feedback loops
- ◉ Unclear or constantly changing requirements: Results in rework, frustration, quality issues, lack of focus
- ◉ Slow or ineffective communication: Results in delays, frustrations, and poor communication to stakeholders which can impact IT's reputation in the organization
- ◉ Partially done work: Does not add value to the customer or allow team to learn from work
- ◉ Defects and quality issues: Results in rework, abandoned work, and poor customer satisfaction



# BUILD QUALITY IN

- ◉ Pair programming: Avoid quality issues by combining the skills and experience of two developers instead of one
- ◉ Test-driven development: Writing criteria for code before writing the code to ensure it meets business requirements
- ◉ Incremental development and constant feedback
- ◉ Minimize wait states: Reduce context switching, knowledge gaps, and lack of focus
- ◉ Automation any tedious, manual process or any process prone to human error

# CREATE KNOWLEDGE

- ◉ Pair programming
- ◉ Code reviews
- ◉ Documentation
- ◉ Wiki - to let the knowledge base build up incrementally
- ◉ Thoroughly commented code
- ◉ Knowledge sharing sessions
- ◉ Training
- ◉ Use tools to manage requirements or user stories

# DEFER COMMITMENT

- ◉ Not plan (in excessive detail) for months in advance
- ◉ Not commit to ideas or projects without a full understanding of the business requirements
- ◉ Constantly be collecting and analyzing information regarding any important decisions

# DELIVER FAST

- ◉ Thinking too far in advance about future requirements
- ◉ Blockers that aren't responded to with urgency
- ◉ Over-engineering solutions and business requirements

# RESPECT FOR PEOPLE

- ◉ Communicating proactively and effectively
- ◉ Encouraging healthy conflict
- ◉ Surfacing any work-related issues as a team
- ◉ Empowering each other to do their best work

# ADVANTAGES OF LSD

LSD has proved to improve software development in following ways :

- ◉ LSD removes the unnecessary process stages when designing a software so that it act as a time saver as simplifies the development process.
- ◉ With focus on MVP, Lean Software Development prioritizes essential functions so this removes the risk of spending time on valueless builds.
- ◉ It increases involvement power of your team as more and more members participate due to which the overall workflow becomes optimized and losses gets reduced.

# THANK YOU