Agile, XP and Scrum

Topics covered

- ✧ Agile methods
- ✧ Agile development techniques
- ✧ Agile project management

Agile development

- Program specification, design and implementation are inter-leaved
- The system is developed as a series of versions or increments with stakeholders involved in version specification and evaluation
- Frequent delivery of new versions for evaluation
- Extensive tool support (e.g. automated testing tools) used to support development.
- Minimal documentation – focus on working code

Agile Manifesto – Core Values

- ✧ 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 Manifesto – 12 Principles

The agile manifesto also identifies 12 principles:

1. *Customer satisfaction by early and continuous delivery of valuable software.*
2. *Welcome changing requirements, even in late development.*
3. *Deliver working software frequently (weeks rather than months).*
4. *Close, daily cooperation between business people and developers.*
5. *Projects are built around motivated individuals, who should be trusted.*

Agile Manifesto – 12 Principles

6. *Face-to-face conversation is the best form of communication (co-location).*
7. *Working software is the primary measure of progress.*
8. *Sustainable development, able to maintain a constant pace.*
9. *Continuous attention to technical excellence and good design.*
10. *Simplicity - the art of maximizing the amount of work not done - is essential.*
11. *Best architectures, requirements, and designs emerge from self-organizing teams.*
12. *Regularly, the team reflects on how to become more effective, and adjusts accordingly.*

Agile principles and organizational practice

Principle	Practice
Customer involvement	This depends on having a customer who is willing and able to spend time with the development team and who can represent all system stakeholders. Often, customer representatives have other demands on their time and cannot play a full part in the software development. Where there are external stakeholders, such as regulators, it is difficult to represent their views to the agile team.
Embrace change	Prioritizing changes can be extremely difficult, especially in systems for which there are many stakeholders. Typically, each stakeholder gives different priorities to different changes.
Incremental delivery	Rapid iterations and short-term planning for development does not always fit in with the longer-term planning cycles of business planning and marketing. Marketing managers may need to know what product features several months in advance to prepare an effective marketing campaign.

Agile principles and organizational practice

Principle	Practice
Maintain simplicity	Under pressure from delivery schedules, team members may not have time to carry out desirable system simplifications.
People not process	Individual team members may not have suitable personalities for the intense involvement that is typical of agile methods, and therefore may not interact well with other team members.

Agile method applicability

- Product development where a software company is developing a small or medium-sized product for sale.
  - Virtually all software products and apps are now developed using an agile approach
- Custom system development within an organization, where there is a clear commitment from the customer to become involved in the development process and where there are few external rules and regulations that affect the software.

Agile Development Practices

Extreme programming

- A very influential agile method, developed in the late 1990s, that introduced a range of agile development techniques.
- Extreme Programming (XP) takes an ‘extreme’ approach to iterative development.
  - New versions may be built several times per day;
  - Increments are delivered to customers every 2 weeks;
  - All tests must be run for every build and the build is only accepted if tests run successfully.

Influential XP practices

- Extreme programming has a technical focus and is not easy to integrate with management practice in most organizations.
- Key practices
  - User stories for specification
  - Refactoring
  - Test-first development
  - Test-driven development
  - Pair programming

User stories for requirements

- In XP, a customer or user is part of the XP team and is responsible for making decisions on requirements.
- User requirements are expressed as user stories or scenarios.
- These are written on cards and the development team break them down into implementation tasks. These tasks are the basis of schedule and cost estimates.

A ‘prescribing medication’ story

Prescribing medication

The record of the patient must be open for input. Click on the medication field and select either ‘current medication’, ‘new medication’ or ‘formulary’.

If you select ‘current medication’, you will be asked to check the dose; If you wish to change the dose, enter the new dose then confirm the prescription.

If you choose, ‘new medication’, the system assumes that you know which medication you wish to prescribe. Type the first few letters of the drug name. You will then see a list of possible drugs starting with these letters. Choose the required medication. You will then be asked to check that the medication you have selected is correct. Enter the dose then confirm the prescription.

If you choose ‘formulary’, you will be presented with a search box for the approved formulary. Search for the drug required then select it. You will then be asked to check that the medication you have selected is correct. Enter the dose then confirm the prescription.

In all cases, the system will check that the dose is within the approved range and will ask you to change it if it is outside the range of recommended doses.

After you have confirmed the prescription, it will be displayed for checking. Either click ‘OK’ or ‘Change’. If you click ‘OK’, your prescription will be recorded on the audit database. If you click ‘Change’, you reenter the ‘Prescribing medication’ process.

Examples of task cards for prescribing medication

Task 1: Change dose of prescribed drug

Task 2: Formulary selection

Task 3: Dose checking

Dose checking is a safety precaution to check that the doctor has not prescribed a dangerously small or large dose.
Using the formulary id for the generic drug name, lookup the formulary and retrieve the recommended maximum and minimum dose.
Check the prescribed dose against the minimum and maximum. If outside the range, issue an error message saying that the dose is too high or too low.
If within the range, enable the 'Confirm' button.

Refactoring

- Programming team look for possible software improvements and make these improvements even where there is no immediate need for them.
- This improves the understandability of the software and so reduces the need for documentation.
- Changes are easier to make because the code is well-structured and clear.
- However, some changes requires architecture refactoring and this is much more expensive.

Test-first development

- Testing is central to XP and XP has developed an approach where the program is tested after every change has been made.
- XP testing features:
  - Test-first development.
  - Incremental test development from scenarios.
  - User involvement in test development and validation.

Test-driven development

- Writing tests before code, clarifies the requirements to be implemented.
  - Writing tests first helps you understand exactly what the code should do.
- Tests are written as programs rather than data so that they can be executed automatically. The test includes a check that it has executed correctly.
  - Usually relies on a testing framework such as Junit.

Pair programming

- Pair programming involves programmers working in pairs, developing code together.
- This helps develop common ownership of code and spreads knowledge across the team.
- It serves as an informal review process as each line of code is looked at by more than 1 person.

Pair programming

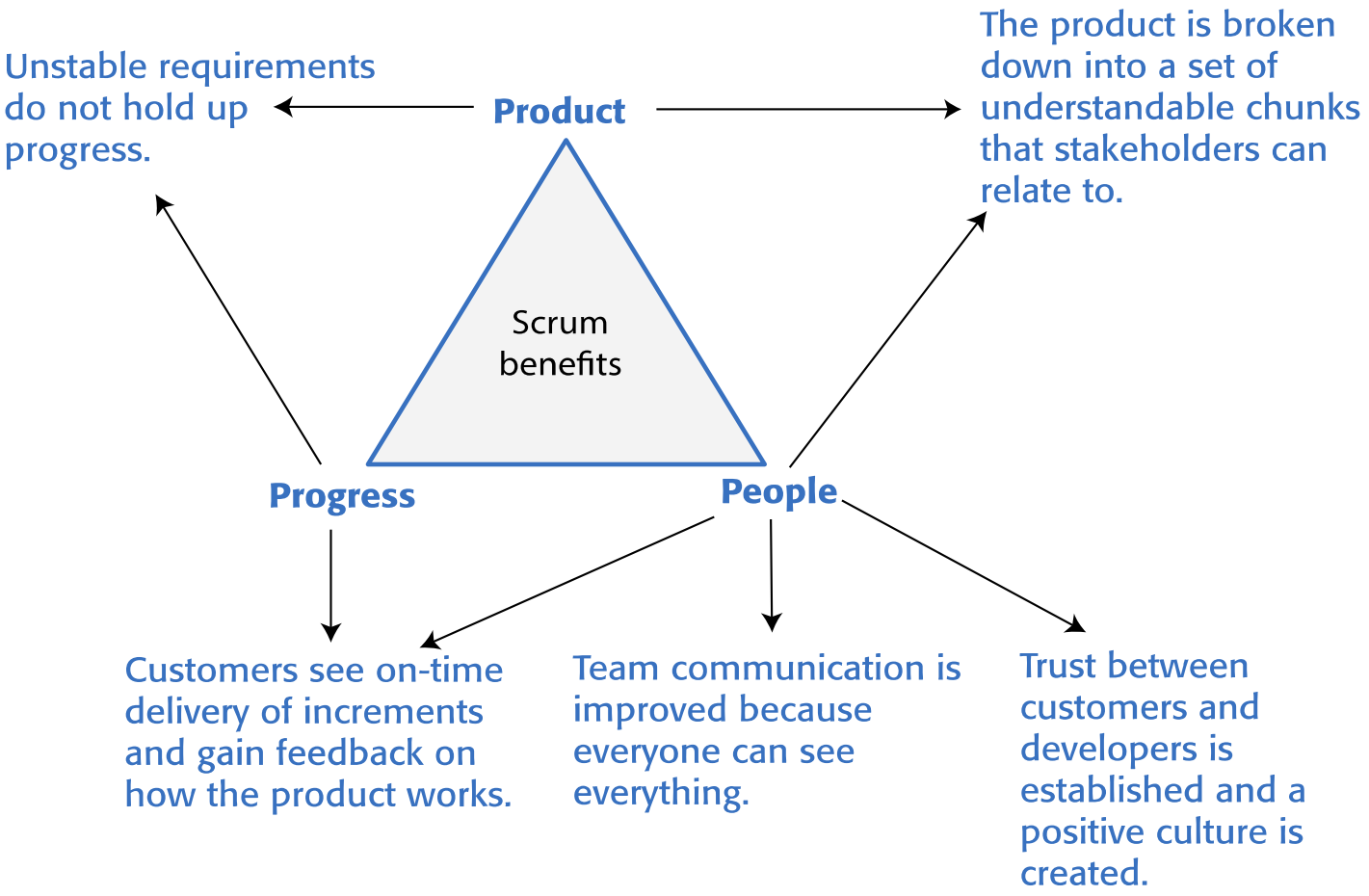
- In pair programming, programmers sit together at the same computer to develop the software.
- Pairs are created dynamically so that all team members work with each other during the development process.
- The sharing of knowledge that happens during pair programming is very important as it reduces the overall risks to a project when team members leave.

Agile Project Management

Scrum

- Scrum is an agile method that focuses on managing iterative development rather than specific agile practices.
- There are three phases in Scrum.
  - The initial phase is an outline planning phase where you establish the general objectives for the project and design the software architecture.
  - This is followed by a series of sprint cycles, where each cycle develops an increment of the system.
  - The project closure phase wraps up the project, completes required documentation such as system help frames and user manuals and assesses the lessons learned from the project.
-

Scrum benefits



Scrum terminology (a)

Scrum term	Definition
Development team	A self-organizing group of software developers, which should be no more than 7 people. They are responsible for developing the software and other essential project documents.
Potentially shippable product increment	In Agile and Scrum frameworks, a "Potentially Shippable Product Increment" (PSPI) refers to a completed, tested, and usable version of the product that could potentially be released to customers at the end of each Sprint.
Product backlog	This is a list of ‘to-do’ items that the Scrum team must tackle. These may include feature definitions, software requirements, or user stories.
Product owner	An individual (or possibly a small group) whose job is to identify product features or requirements, prioritize these for development and continuously review the product backlog to ensure that the project continues to meet critical business needs.

Scrum terminology (b)

Scrum term	Definition
Scrum	A daily meeting of the Scrum team that reviews progress and prioritizes work to be done that day. Ideally, this should be a short face-to-face meeting that includes the whole team.
ScrumMaster	The ScrumMaster is responsible for ensuring that the Scrum process is followed and guides the team in the effective use of Scrum. He or she is responsible for interfacing with the rest of the company and for ensuring that the Scrum team is not diverted by outside interference.
Sprint	A development iteration. Sprints are usually 2-4 weeks long.
Sprint Review	A meeting held at the end of the Sprint to inspect the product increment and gather feedback from stakeholders.
Sprint Retrospective	A meeting for the Scrum team to reflect on the Sprint and identify improvements for the next Sprint.

The Scrum Process

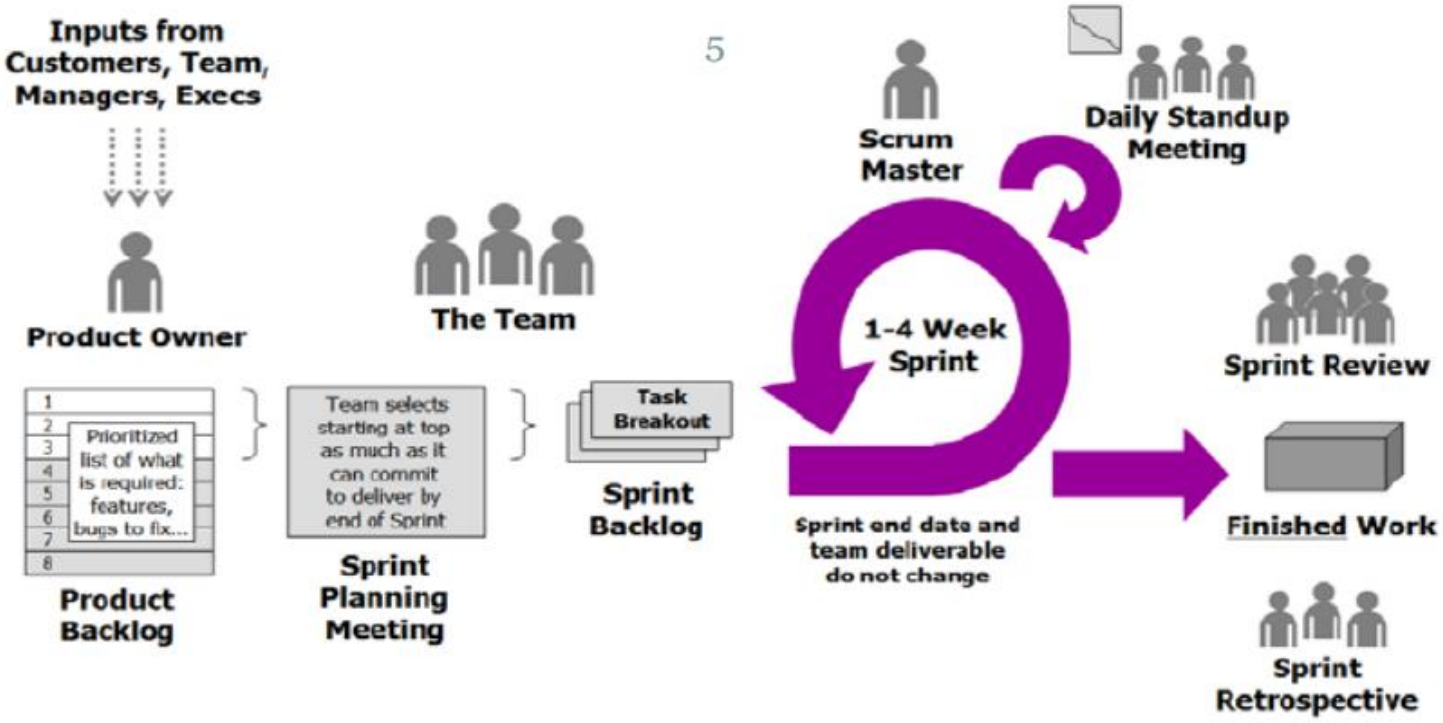


Figure 1. Scrum

The Scrum Sprint Cycle

- Sprints are fixed length, normally 2–4 weeks.
- The starting point for planning is the product backlog, which is the list of work to be done on the project.
- The selection phase involves all of the project team who work with the customer to select the features and functionality from the product backlog to be developed during the sprint.

The Scrum Sprint Cycle

- Once these are agreed, the team organize themselves to develop the software.
- During this stage the team is isolated from the customer and the organization, with all communications channelled through the so-called ‘Scrum master’.
- The role of the Scrum master is to protect the development team from external distractions.
- At the end of the sprint, the work done is reviewed and presented to stakeholders. The next sprint cycle then begins.

Scrum Roles

- In Scrum, there are three primary roles:
- 1. The Product Owner
- 2. Team Members
- 3. The ScrumMaster

Product Owner

- The Product Owner is responsible for taking all the inputs into what the product should be – from the customer or end-user of the product, as well as from Team Members and stakeholders – and translating them into a product vision.
- In some cases, the Product Owner and the customer are one and the same.
- Product owner is responsible for product backlog, manages release plan, adjusts priorities based on ROI and business value, accepts or rejects work results.

Team Members

- Team Members build the product that the customer is going to consume: the software, the website, or whatever it may be.
- The team in Scrum is typically five to ten people, although teams as large as 15 and as small as 3 commonly report benefits.
- The team should include all the expertise necessary to deliver the finished work – so, for example, the team for a software project might include roles (programmers, interface designers, testers, marketers, and researchers).

Scrum Master

- The Scrum Master is tasked with doing whatever is necessary to help the team be successful.
- The Scrum Master is not the manager of the team; he or she serves the team, by helping remove blocks to the team's success, facilitating meetings, and supporting the practice of Scrum.
- Some teams will have someone dedicated fully to the role of Scrum Master, while others will have a team member play this role (carrying a lighter load of regular work when they do so).

Scrum Master

- Great Scrum Masters have come from all backgrounds and disciplines: Project Management, Engineering, Design, Testing.
- The Scrum Master and the Product Owner probably shouldn't be the same individual; at times, the Scrum Master may be called upon to push back on the Product Owner (for example, if they try to introduce new requirements in the middle of a Sprint).

Product Backlog

- The product backlog is a list of what needs to be done to complete the development of the product.
- The items on this list are called product backlog items (PBIs).
- The product backlog may include a variety of different items such as product features to be implemented, user requests, essential development activities and desirable engineering improvements.
- The product backlog should always be prioritized so that the items that be implemented first are at the top of the list.

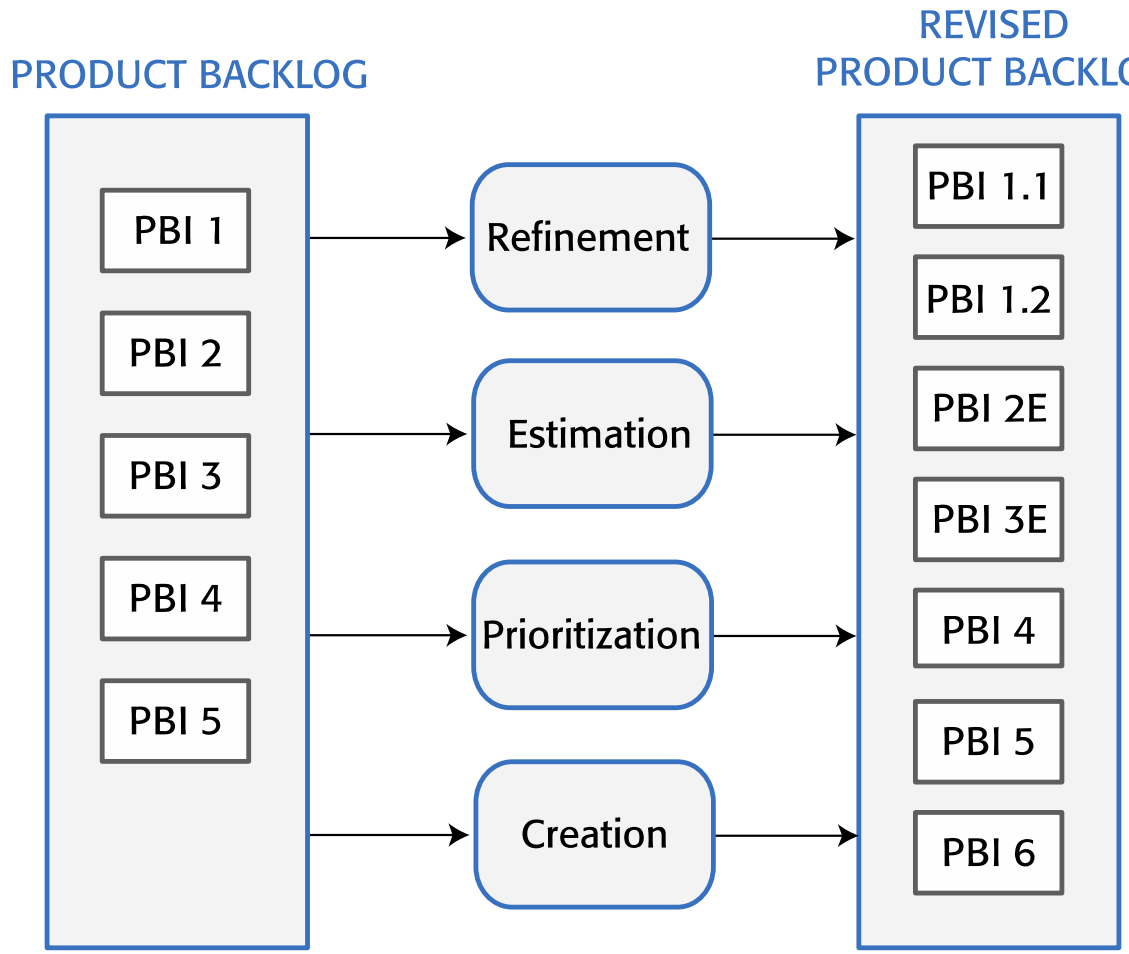
Product Backlogs Entries (Examples)

- *As a teacher, I want to be able to configure the group of tools that are available to individual classes. (feature)*
- *As a parent, I want to be able to view my childrens' work and the assessments made by their teachers. (feature)*
- *Establish criteria for the assessment of open source software that might be used as a basis for parts of this system. (development activity)*
- *Refactor user interface code to improve understandability and performance. (engineering improvement)*
- *Implement encryption for all personal user data. (engineering improvement)*

Product Backlog Item States (Typical)

- ***Ready for consideration***
  These are high-level ideas and feature descriptions that will be considered for inclusion in the product. They are tentative so may radically change or may not be included in the final product.
- ***Ready for refinement***
  The team has agreed that this is an important item that should be implemented as part of the current development. There is a reasonably clear definition of what is required. However, work is needed to understand and refine the item.
- ***Ready for implementation***
  The PBI has enough detail for the team to estimate the effort involved and to implement the item. Dependencies on other items have been identified.

Product Backlog Activities



Product Backlog (Example)

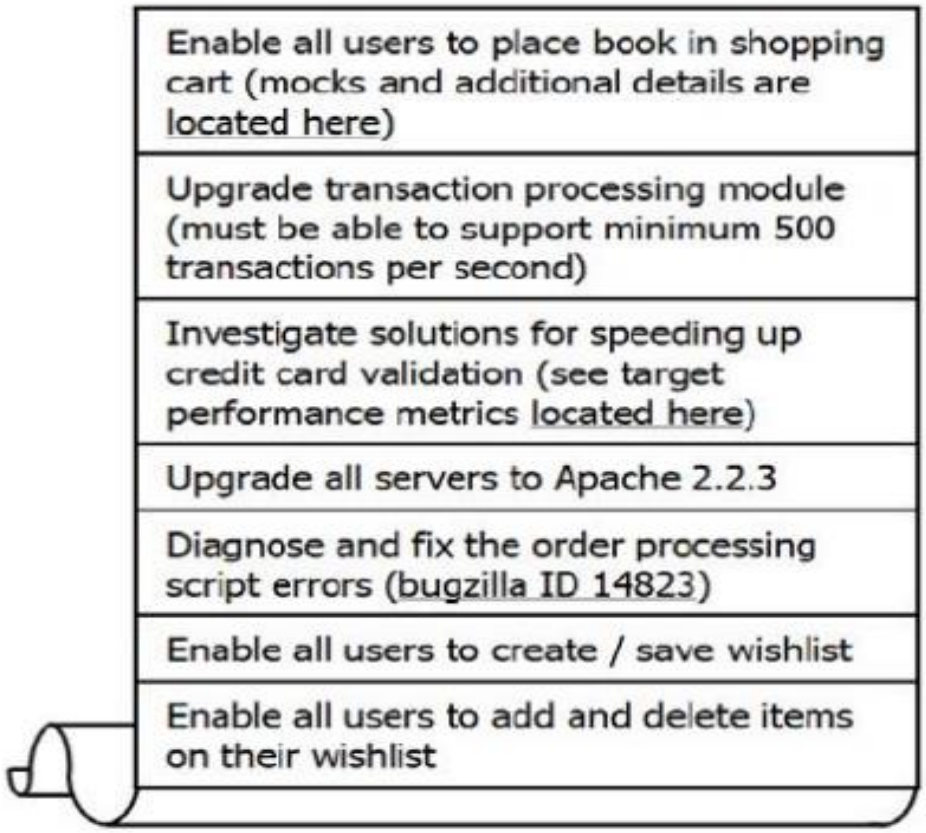


Figure 2. The Product Backlog

Sprint Planning Meeting

- In the first part of the Sprint Planning Meeting, the Product Owner and Scrum Team (with facilitation from the Scrum Master) review the Product Backlog, discussing the goals and context for the items on the Backlog, and providing the Scrum Team with insight into the Product Owner's thinking.
- In the second part of the meeting, the Scrum Team selects the items from the Product Backlog to commit to complete by the end of the Sprint, starting at the top of the Product Backlog (in others words, starting with the items that are the highest priority for the Product Owner) and working down the list in order.

;

Sprint Planning Meeting

- This is one of the key practices in Scrum: the team decides how much work they will commit to complete, rather than having it assigned to them by the Product Owner. This makes for a much more reliable commitment
- The Sprint Planning meeting will often last a number of hours
- the team is making a very serious commitment to complete the work, and this commitment requires careful thought to be successful.

Sprint Planning Meeting

- The team will begin by estimating how much time each member has for Sprint-related work – in other words, their average workday minus the time they spend doing things like critical bug-fixes and other maintenance, attending meetings, doing email, taking lunch breaks, and so on. For most people this works out to 4-6 hours of time per day available for Sprint-related work. (Figure 3.)

Sprint Planning Meeting

Sprint Length	2 weeks
Workdays During Sprint	10 days

Team Member	Available Days During Sprint*	Available Hours Per Day	Total Available Hours During Sprint
Tracy	9 days	4 hours	36 hours [=9 x 4]
Sanjay	10 days	5 hours	50 hours
Phillip	10 days	4 hours	40 hours
Jing	8 days	5 hours	40 hours

*Net of vacation, other days out of the office

Figure 3. Estimating Available Hours

Sprint backlog

- Once the time available is determined, the team starts with the first item on the Product Backlog
- In other words, the Product Owner's highest priority item – and working together, breaks it down into individual tasks, which are recorded in a document called the Sprint Backlog (Figure 4).
- At the end of the meeting, the team will have produced a list of all the tasks, and for each task who has signed up to complete it and how much time they estimate it will take (typically in hours or fractions of a day).

Sprint backlog

Backlog Item	Task	Owner	Initial Time Estimate
Upgrade transaction processing module	Configure database and space IDs for Trac	Sanjay	4 hours
	Use test data to tune the learning and action model	Jing	2 hours
	Setup a cart server code to run as apache server	Philip	3 hours
	Implement pre-Login Handler	Tracy	3 hours
Investigate solutions for speeding up credit card validation	Merge DCP code and complete layer-level tests	Jing	5 hours
	Complete machine order for pRank	Jing	4 hours
	Change DCP and reader to use pRank http API	Tracy	3 hours

Figure 4. Sprint Backlog

Scrum Change

- One of the key pillars of Scrum is that once the Scrum Team makes its commitment, the Product Owner cannot add new requests during the course of the Sprint.
  - This means that even if halfway through the Sprint the Product Owner decides that they want to add something new, he or she cannot make changes until the start of the next Sprint.
  - If an external circumstance appears that significantly changes priorities, and means the team would be wasting its time if it continued working, the Product Owner can terminate the Sprint.
    - This means the team stops all the work they are doing, and starts over with a Sprint Planning meeting, and so forth.

Daily Standup Meeting

- Once the Sprint has started, the Scrum Team engages in another of the key Scrum practices: The Daily Stand-Up Meeting.
- This is a short (15 minute) meeting that happens every workday at an appointed time, and everyone on the Scrum Team attends; in order to ensure it stays brief, everyone stands (hence “Stand-Up Meeting”).
- It’s the team’s opportunity to report to itself on progress and obstacles.

Daily Standup Meeting

- One by one, each member of the team reports just three things to the other members of the team:
  1. **What they were able to get done since the last meeting.**
  2. **What they're aiming to get done by the next meeting.**
  3. **Any blocks or obstacles that are in their way.**
- There's no discussion during the Daily Stand-Up Meeting, just the reporting of the three key pieces of information; if discussion is required, it takes place right after the meeting.
- The Scrum Master makes note of the blocks, and then helps team members to resolve them after the meeting.

Sprint status

- One of the core tenets of Scrum is that the duration of the Sprint is never extended – it ends on the assigned date regardless of whether the team has completed the work it committed to or not.
- If the team has not completed their Sprint Goal, the remaining tasks are re-prioritized and the issue is brought up the sprint review and retrospective meeting .

Sprint Review

- At the end of each sprint, there is a review meeting, which involves the whole team. This meeting:
  - reviews whether or not the sprint has met its goal.
  - sets out any new problems and issues that have emerged during the sprint.
  - is a way for a team to reflect on how they can improve the way they work.
- The product owner has the ultimate authority to decide whether or not the goal of the print has been achieved. They should confirm that the implementation of the selected product backlog items is complete.

Sprint Review

- The sprint review should include a process review, in which the team reflects on its own way of working and how Scrum has been used.
  - The aim is to identify ways to improve and to discuss how to use Scrum more productively.

Sprint Retrospective

- Following the Sprint Review, the team gets together for the Sprint Retrospective.
- The focus of this meeting is process improvement rather than productivity improvement.
- It's an opportunity for the team to discuss what's working and what's not working, and agree on changes to try.
- Three questions are asked off each participant:
  - **What went well for you in this sprint?**
  - **What went bad?**
  - **What can be improved?**

Starting the Next Sprint

- Following the Sprint Review Meeting, the Product Owner takes all the input, as well as all new priorities that have appeared during the Sprint, and incorporates them into the Product Backlog; new items are added, and existing ones are modified, reordered, or deleted.
- Once this updating of the Product Backlog is complete, the cycle is ready to begin all over again, with the next Sprint Planning Meeting.
- One practice many teams find useful is to hold a Prioritization Meeting toward the end of each Sprint, to review the Product Backlog for the upcoming Sprint with the Product Owner.

Contrast Scrum project management with traditional plan-based methods.

1- Planning allocation of people to projects:

- **Scrum**
- Scrum handles people allocation informally. Team members 'bid' for features from the product backlog to implement if they think that their expertise is appropriate. Alternatively, the tasks can be allocated by the Scrum master. There is no formal mechanism in Scrum for planning for project members with very specific expertise to be temporarily allocated to a team. This need must be identified by the Scrum master and he or she has to discuss how the expertise can be made available.
- **Plan-based development**
- A project plan is used to identify the parts of the system to be delivered and these are specified in the requirements document. The expertise required for each part can then be identified and the allocation of people to projects planned on that basis.

Contrast Scrum project management with traditional plan-based methods.

2- Estimating project costs:

- **Scrum**
- Project costs are estimated based on the required delivery date for the software and people working in the Scrum team. The functionality of the system is adjusted so that some working system will always be delivered for the original cost estimation. Of course, this may not be adequate for the customer and they have to become involved in rescheduling the delivery of the system.
- **Plan-based development**
- Project costs are based on an analysis of the functionality specified in the requirements document as well as the non-functional requirements of the system. They may be adjusted to reflect team size and delivery schedule. It is normal for costs to be underestimated and the final project to cost much more than originally estimated. An average cost for team members is assumed.

Contrast Scrum project management with traditional plan-based methods.

3- Maintaining team cohesion:

- **Scrum**
- Team member meet daily either face to face or electronically. Extensive informal discussions and communications are encouraged. Team members negotiate work to be done from the project backlog.
- **Plan-based development**
- Team cohesion is the responsibility of the project manager and he or she has to take explicit actions to encourage this. The general approach relies on formal meetings that are relatively infrequent and this does not lead to the development of a cohesive team.