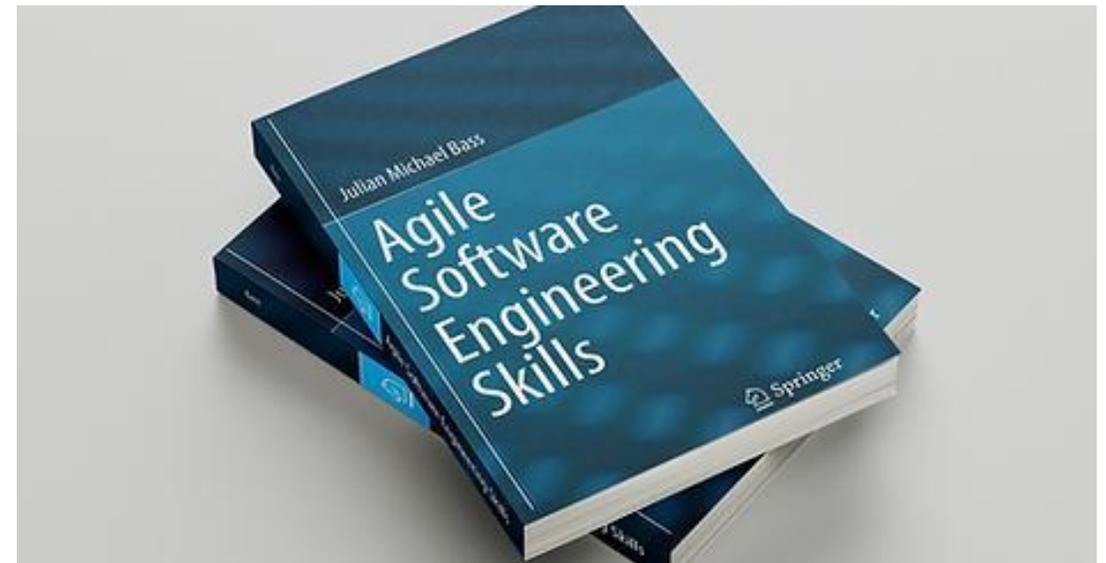


Agile Software Engineering Skills

Agile Ceremonies

Chapter 13

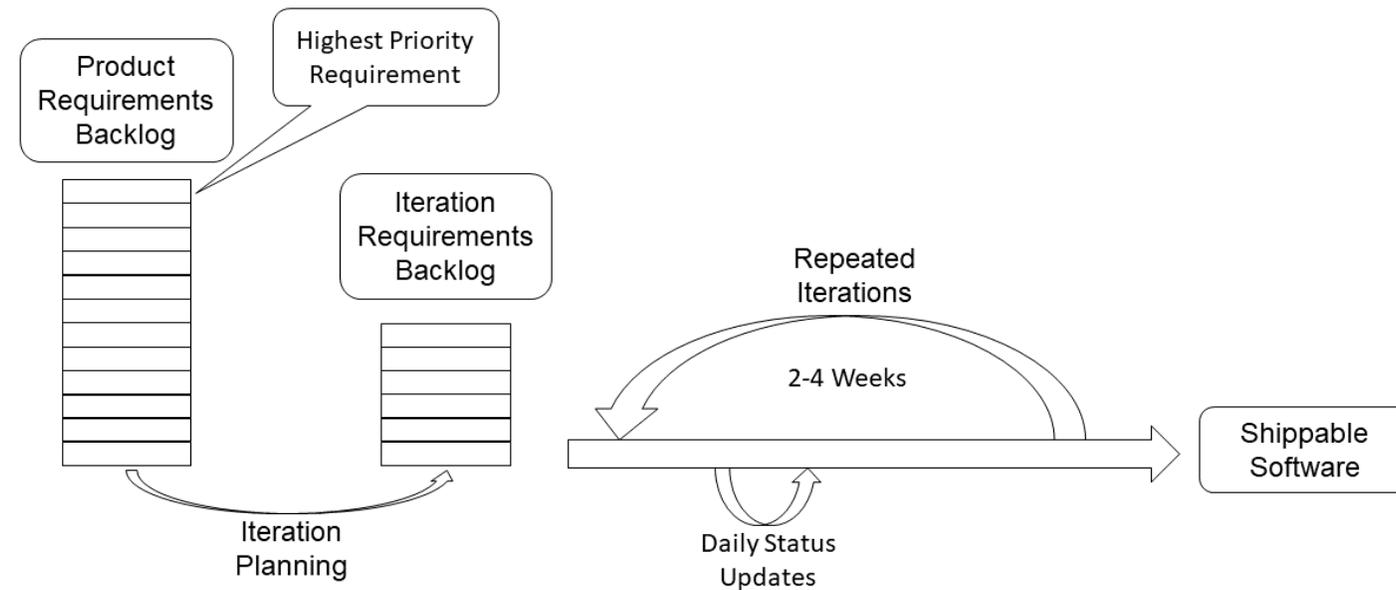
Julian M. Bass



Introduction

- Ceremonies are the group collaboration activities
- Iterations start with planning
- Each iteration has a review at the end to get customer feedback

Introduction



Contents

- Iteration Planning
- Coordination Meetings
- Customer Demonstrations
- Pair Programming
- Test-Driven Development
- Specialist Agile Ceremonies

Iteration Planning

- Deciding what to work on next
- Conducted at the start of each iteration
- Comprises five tasks
 - prioritisation of requirements
 - break-up requirements into technical tasks
 - estimation of technical tasks and consequently requirements
 - work item assignment within the team

Iteration Planning

- **Prioritisation**
 - Product owners prioritise requirements
- **Features and Technical Tasks**
 - Break each feature (user story) into technical tasks
 - Creating a list of smaller work items for each requirement

Iteration Planning

- Summary process, as follows
 - Select highest priority user story from the backlog
 - Discuss the purpose and scope of the user story
 - The product owner answers questions about the user story
 - The discussion is complete, once all questions have been answered
 - The user story is broken-up into constituent technical tasks
 - Depending upon the application domain, think about user interface tasks, application logic and data storage tasks as separate items
 - Repeat the process for the next high priority user story in the product backlog

Iteration Planning

- Estimation
 - Need to know how many features we can fit into an iteration
 - Too few, team is under utilised
 - Too many, difficult to maintain sustainable pace
 - Two main approaches
 - Story points
 - Based on Fibonacci sequence 1, 3, 5, 8 and 13
 - T-Shirt sizing
 - Small, medium, large, extra large

Iteration Planning

- Planning poker technique
 - Take each technical task in turn, the first round of voting starts
 - Discuss each technical task, if necessary
 - Team members write down (secretly) their estimates for the work item
 - When everyone has finished writing, team members reveal their votes for the tasks
 - Look at the story points assigned, see if there is close consensus (in novice teams or a new application domain, close consensus is unlikely)

Iteration Planning

- Planning poker technique (Cont.)
 - If there is consensus, on the story point allocation, move on to the next technical task
 - If there is not consensus, constructively discuss the highest and lowest story point estimates, try to understand why someone thought it was a larger or smaller task
 - Following this discussion move into a second round of voting
 - Continue rounds of voting and discussion until consensus emerges around the story point value for a task
 - Then, move on to the next technical task or user story

Iteration Planning

- Task Assignment
 - Decide who, in the team, is going to tackle each task
 - People volunteer for tasks
 - Self-organising teams develop a sense of shared commitment to group outcomes
 - Unpopular tasks tend to get shared around

Coordination Meetings

- Daily stand-up meetings
 - Coordination meeting involving everyone in the team
 - Everyone finds out what is going on
 - Everyone answers three questions:
 - What have I been doing, since the last stand-up?
 - What will I be doing, between now and the next stand-up?
 - Are there any impediments preventing me from making progress?

Coordination Meetings

- Virtual stand-up meetings
 - On-line stand-up meetings
 - Where team members are working remotely or geographically distributed
 - Hybrid (some people on-line, some people physically attending) best avoided

Coordination Meetings

- Kanban Boards
 - Make visual the teams efforts towards project goals, (see Chap 10)
 - Usually consists of three columns
 - To Do, Doing and Done
 - Tasks start off in the To Do column
 - As project progresses, tasks are moved to the Done column

Customer Demonstrations

- Demonstrate working code at the end of an iteration
- During the customer demonstration
 - Introduce the purpose of the meeting
 - Review the requirements you were supposed to implement
 - Demonstrate each new feature of the software
 - Review any requirements that you were unable to implement for any reason
 - Collect and carefully record any feedback from the product owner or client

Customer Demonstrations

- Retrospectives
 - Opportunity to learn from each iteration
 - Everyone in the team writes down and shares
 - Three things that worked well in the previous iteration
 - Three things that, as a team, you could do better
 - Then establish consensus on
 - Three improvement actions for the next sprint

Pair Programming

- Two developers work together in a pilot, co-pilot configuration
 - One developer is more focused on low-level syntax
 - This developer has a keyboard and is actually typing source code
 - The second developer is considering higher-level structure and readability
 - Focus on source code quality and acceptance testing

Test-Driven Development

- Automated tests are written before the code itself
- A cycle of activities
 - Write a test
 - Make it run
 - Make it right

Specialist Agile Ceremonies

- Spikes
 - Occur where the estimate, for a requirement or technical task under development, proves to be inaccurate
 - Usually means some previously hidden complexity has emerged
 - Might choose to park the story for a future sprint
 - Better might be to consider adding resources to that story
 - Product owner might reduce the priority of some other activity, so that we can resolve the spike

Specialist Agile Ceremonies

- **Swarm Programming**
 - More than two developers work together
 - Useful to tackle some new task or technology
 - Or, where progress is blocked by one particular problem
 - Use swarm programming to address a high-priority spike
- **Mob Programming**
 - Whole team works together all the time on same code

Exercises

- Exercises 13.1 and 13.6 encourage creation of a learning journal
- Exercise 13.2 Explores iteration planning
- Exercise 13.3 Perform a stand-up status meeting
- Exercise 13.4 Perform a customer demonstration
- Exercise 13.5 Perform a retrospective

Summary

- Iteration Planning
 - Estimate tasks and select work items for iteration
- Coordination Meetings
 - Team status meetings, often held daily
- Customer Demonstrations
 - Show customer working code to get feedback
 - Retrospective
 - Learn from previous iteration and set improvement goals for next one

Summary

- **Pair Programming**
 - Pilot, co-pilot programming configuration
- **Test-Driven Development**
 - Create tests first, then develop code to pass tests
- **Specialist Agile Ceremonies**
 - Spikes, when things go wrong with a story during an iteration
 - Swarm Programming, work as a team on source code when needed
 - Mob programming, work as a team on source code always