University of Salford MANCHESTER

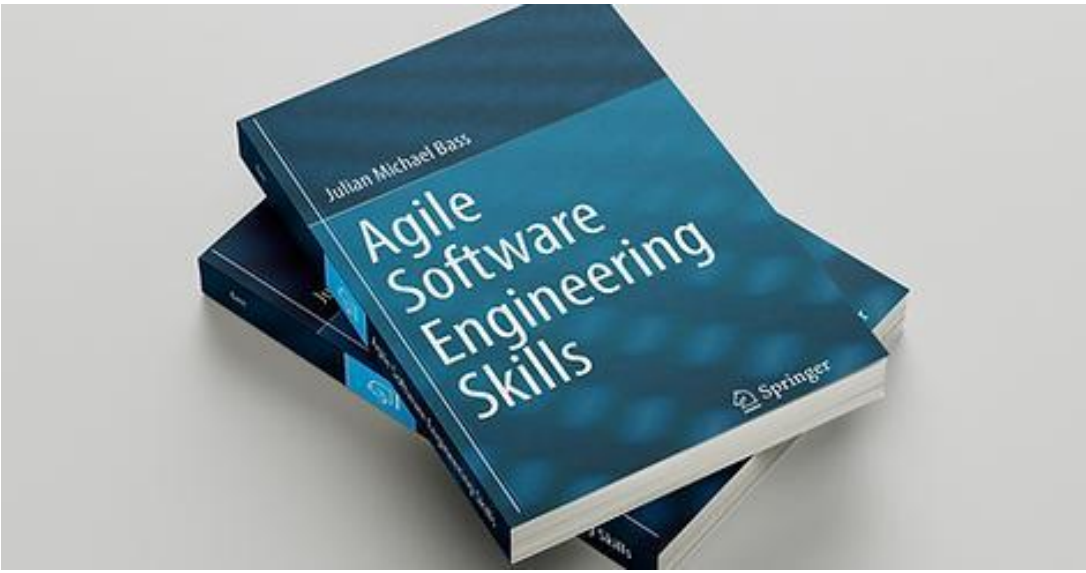
REDocelot

Agile Software Engineering Skills

Development

Chapter 10

Julian M. Bass



University of Salford MANCHESTER

REDocelot

Introduction

- We create computer programs during software development iterations
- Software source code files are development artefacts
- Other artefacts are also produced
  - For planning and releases, for instance
  - Bass, J. M. (2016). Artefacts and agile method tailoring in large-scale offshore software development programmes. *Information and Software Technology*, *75*, 1–16. https://doi.org/10.1016/j.infsof.2016.03.001
- Artefacts enable communication between stakeholders

University of Salford MANCHESTER

REDocelot

Contents

- Planning Artefacts
- Iteration Artefacts
- Feature Artefacts
- Release Artefacts

University of Salford MANCHESTER

REDocelot

Planning Artefacts

- Kanban Boards
  - Physical, or virtual, Kanban boards summarise project status
  - Can comprise three columns
    - To Do, a backlog of work items
    - In Progress, work items under development
    - Done, completed work items
  - Work items move from one column to the next, in status meetings
  - Everyone can see project status

Planning Artefacts

- Product Backlog
  - Some teams like to split the To Do column
    - Iteration backlog
    - Product backlog
  - Prioritised list of requirements (user stories)
  - As project unfolds, product owner re-prioritises requirements
  - Keep highest priority at the top

University of Salford MANCHESTER

REDocelot

Planning Artefacts

- Test Plan
  - Strategy or policy defining how testing achieves required code quality
  - Test plan must be available and followed by everyone
  - Describes testing performed at each development stage
  - Considers the desired level of test coverage
    - Number of code pathways tested
  - Think about the different types of testing you need to perform
    - Unit, regression, load, integration, user acceptance testing…

Iteration Artefacts

- Iteration Backlog
  - Subset of requirements from product backlog
  - Highest priority user stories from product backlog
  - Enough user stories to occupy team for an iteration
    - Depends on estimation of work items

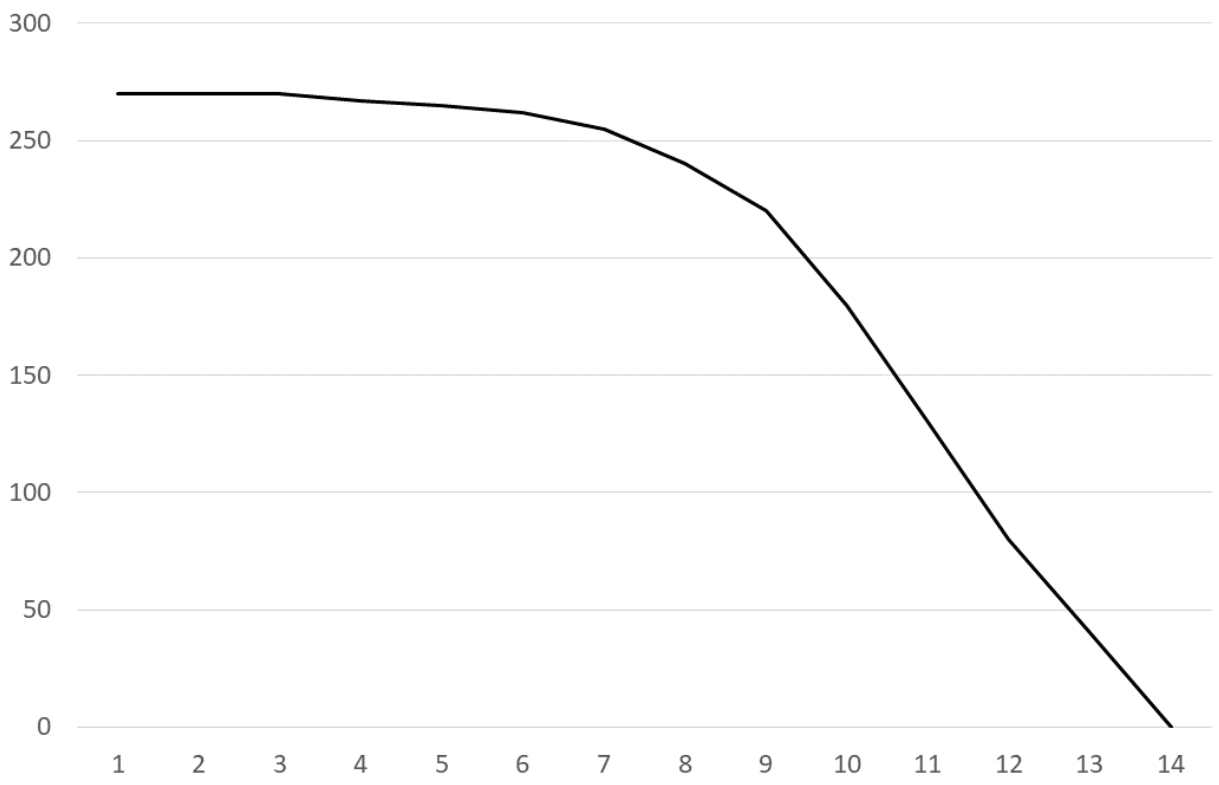
Iteration Artefacts

- User Story Estimates
  - Break user stories into technical tasks required
  - Estimate effort required for each technical task
  - T-shirt sizing approach can be used
    - Small, medium, large, extra large…

Iteration Artefacts

- Burn Down Chart
  - illustrates project progress during an iteration
  - Y-axis represents story points
  - X-axis the number of days in the iteration
  - Stories shown on chart when they are actually completed
  - Chart should accurately reflect progress
    - Story points might get added during iteration, perhaps due to new information
    - All tasks might not get completed

Iteration Artefacts



© Springer Nature Switzerland AG 2022
J. M. Bass, Agile Software Engineering Skills,
https://doi.org/10.1007/978-3-031-05469-3_10

Feature Artefacts

- Prototypes
  - Low fidelity, low cost, visualisation of something
  - Prototypes or mock-ups, of user interfaces for example, can be useful
    - New type of application and little experience in team
    - New customer and uncertain expectations
  - Enables you to gather feedback before (more expensive) development
  - Win commitment for your work from client

University of Salford MANCHESTER

REDocelot

Feature Artefacts

- Source Code
  - Messy code slows you down
  - Elegant code is easily readable by people and machines
    - Use meaningful names
    - Functions (methods) should only do one thing and do it well
    - Good and elegant code does not need comments
    - Good formatting improves communication
    - Use objects to hide data and expose operations
    - Use exceptions to handle errors
    - Clarity, simplicity and density of expression in unit tests
    - Small classes

Feature Artefacts

- Unit Tests
  - Verify specific components of our software system
  - Test classes
    - Constructors that instantiate objects
  - Test methods
    - Check calling and return parameters
  - Test attributes of classes
    - Check variables have been initialised and so on
  - It is more efficient to automate testing, see Chapter 16

Feature Artefacts

- Issues
  - Defects
  - Change requests
    - New feature requests
    - Feature enhancements
  - Teams supporting live systems need to carefully track issues
  - Product owners decide which issues give the most benefit
    - Assess trade-off between costs and benefits
  - Sometimes useful to let clients vote on issue priorities

University of Salford MANCHESTER

REDocelot

Release Artefacts

- Release Code Binaries
  - Code binaries get deployed to a production environment
  - Usually several increments are grouped to form a release
  - Containerisation offers a standardised deployment platform
    - Bundles software applications, libraries and configuration files
  - Container orchestration software manages deployments

University of Salford MANCHESTER

REDocelot

Release Artefacts

- Regression Tests
  - Re-check existing features, when new features are added
  - Re-runs tests to re-evaluate previously tested software features
  - Ensures new features don’t adversely affect previous features
  - This is where automation pays off, little effort just machine time needed

Exercises

- Exercises 10.1 and 10.6 encourage creation of a learning journal
- Exercise 10.2 Covers Kanban board creation
- Exercise 10.3 Explores Test plan creation
- Exercise 10.4 Covers Estimation
- Exercise 10.5 Creates a burn down chart

University of Salford MANCHESTER

REDocelot

Summary

- Planning Artefacts
  - Kanban board – visualise project status
  - Product backlog – prioritised list of requirements
  - Test plan – testing strategy
- Iteration Artefacts
  - Iteration backlog – high priority requirements for current iteration
  - User story estimation – effort required to complete each user story
  - Burn down chart – diagram of completed tasks against time

University of Salford MANCHESTER

REDocelot

Summary

- Feature Artefacts
  - Prototypes – mock-ups for gaining feedback
  - Source code – computer programmes created during development
  - Unit tests – verify code works as expected
  - Issues – defects and feature enhancements
- Release Artefacts
  - Release binaries – code and scripts for production deployment
  - Regression tests – re-test existing features when new features added