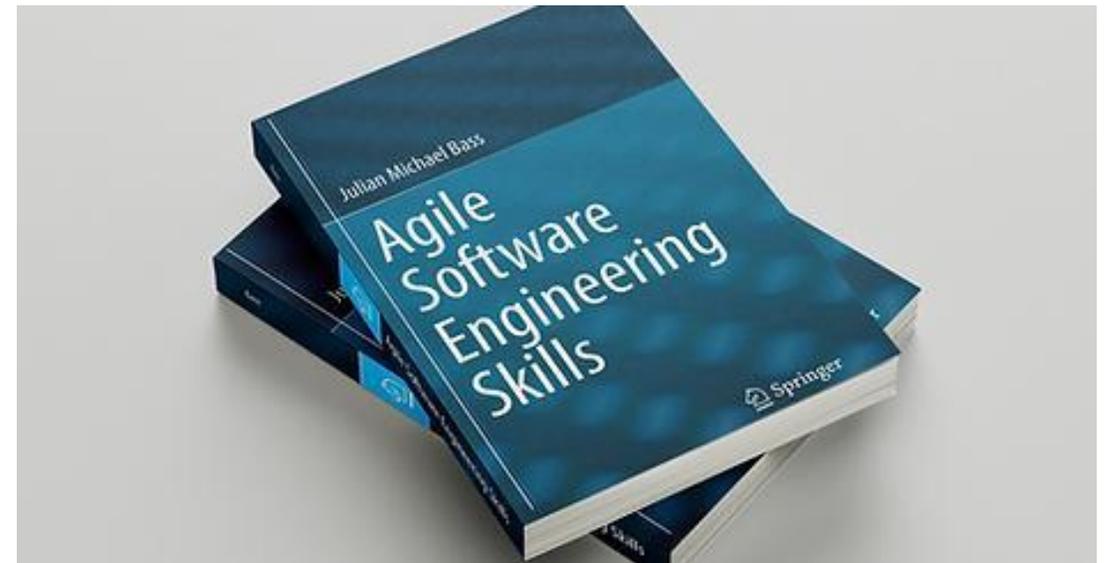


Agile Software Engineering Skills

Lean
Chapter 14
Julian M. Bass



Introduction

- We will use a lean approach
 - Treat each user story or work item as an artefact
 - Flowing through the various stages of a development process
- Focus on concepts such as value, waste, speed, people, knowledge and quality
- Take holistic view of the development process
 - Concentrating on maximising the efficient flow of work items

Introduction

- Lean start-up model
 - Approach to starting a technology company
 - Using revenue to support growth
- We can apply Lean ideas along-side agile methods
- Lean emerged from just-in-time (or smart) manufacturing movement
 - Ideas about responsive production, low inventories and high quality

Introduction

- Seven principles
 - Eliminate Waste, waste is anything that does not add value
 - Build quality in, take steps to avoid creating defects
 - Create knowledge, learn about customer needs in order to fulfil them
 - Defer commitment, experiment with alternatives, only make irreversible decisions when necessary
 - Deliver fast, delays are often expensive
 - Optimise the whole, efficient flow of requirements into products
 - Respect people, support people and build expertise

Contents

- Respecting People
- Create knowledge
- Build quality in
- Value
- Waste
- Speed
- Lean Start-up

Respecting People

- Coach and mentor staff members
 - To acquire required skills and behaviours
 - Help people to develop
- Provide the resources needed to complete work
- Self-organising team responsible for delivering good quality software
 - Team members maybe happy to exercise current skills
 - Sometimes stretch task that creates opportunity to learn new skills

Create knowledge

- Systematic data collection used to inform empirical decision-making
 - Problem definition, focus on the real problem and not a symptom
 - Situation analysis, collect data, usually measurements of your process, to provide evidence
 - Hypothesis creation, untested explanation based on evidence
 - Experimentation, to test the hypothesis
 - Result verification, analyse and replicate experiments
 - Standardisation, make new approach routine

Build quality in

- Strive for perfection
- Quality assurance has two senses
 - Prevention of defects
 - E.g. pair programming and code reviews
 - Detection of errors
 - E.g. test-driven development and test automation

Build quality in

- Focus on code quality
 - Sensible coding standards
 - Automate as much as possible
 - DevOps (Chapter 21) maybe too much for a student or novice project
 - Good for mature commercial teams to consider
 - Use code quality assessment tools
 - Be alert for code smells

Value

- Focus resources on highest value activities
 - Highest value activities will change over time
- We want to maximise value
 - Identify where value is created
 - Eliminate activities that do not produce value
- Non-Monetary Value
 - Legitimate to think beyond monetary value
 - E.g. disaster recovery, primary value = lives saved

Value

- Value Stream Mapping
 - List stakeholders in process
 - List activities performed
 - Estimate activity values
 - Identify dependencies between activities
 - Use value stream to identifying blockages and inefficiencies in process

Value

- Definition of Done
 - Value criteria for work items to flow from one process stage to the next
 - Takes form of a check list or some set of criteria
 - For example, code might be ready for deployment only if
 - Unit tests have all been passed
 - Code Reviews have been completed and any actions addressed
 - Security Tests the full suite of security tests performed
 - Code quality tests have passed

Waste

- Identify and remove any activity that does not add value
- Sources of waste include
 - Partially done work, e.g. untested code, undeployed code
 - Superfluous features, unnecessary features mean bloated products
 - Rework, having to recreate something of poor quality
 - Handoffs, incomplete work item is passed onto someone else
 - Task Switching, Multi-tasking is distracting
 - Delays, waiting time is undesirable
 - Defects, try to catch and remove defects early

Speed

- Speed is the absence of waste
 - Speed in delivering value to clients
 - Efficient development processes, automated testing and DevOps
- Short delivery cycles increase learning

Speed

- Analysis of queueing theory suggests
 - Even-out the arrival of work
 - Minimise number of things in process
 - Minimise size of things in process
 - Establish a regular cadence
 - Limit work to capacity
 - Use pull scheduling

Speed

- Work in Progress Limits
 - Control the number of work items being processed
 - Policy decision to help manage flow
 - Makes team capacity more visible
 - Sometimes desirable to show a buffer on a Kanban board
 - Items blocked by the WIP limit go in buffer
 - If buffer fills, we can choose to add resources to empty buffer

Speed

- Work Item Variability
 - New features, attractive work and we can manage scope
 - Feature enhancements, high priority when customer-driven
 - Refactoring, difficult to estimate and can be large undertaking, at times
 - Defects, difficult to estimate; periodic iterations focused solely on defect fixing can be a useful tactic
- Teams need to gain experience estimating these different tasks types

Lean Start-up

- Create new technology business
- Reaction against over reliance on acquiring investment
- Use prototypes and mock-ups to assess market reaction
- Attract paying customers
 - Experiment to maximise income

Lean Start-up

- **Bootstrapping**
 - Focuses on generating revenue to grow business
 - Pull the business up by its own “bootstraps”
- **Minimum Viable Product (MVP)**
 - Make tangible the essence of a product or solution
 - Minimum set of features needed to make solution work
 - Identify only the essential features
 - Don't invest time and resources on superfluous features

Lean Start-up

- Pivot
 - Sometimes experiments show that MVP is not energising potential customers
 - Change direction towards a variation of our solution
 - Serve a different market
 - Perform a different function
 - Many technology start-ups go through the pivot experience

Exercises

- Exercises 14.1 and 14.8 encourage creation of a learning journal
- Exercise 14.2 Explores a value stream mapping
- Exercise 14.3 Explores waste
- Exercise 14.4 Explores handling requests
- Exercise 14.5 Explores knowledge gathering
- Exercise 14.6 Explores cycle times
- Exercise 14.7 Explores story test-driven development

Summary

- Seven principles
 - Eliminate Waste, waste is anything that does not add value
 - Build quality in, take steps to avoid creating defects
 - Create knowledge, learn about customer needs in order to fulfil them
 - Defer commitment, experiment with alternatives, only make irreversible decisions when necessary
 - Deliver fast, delays are often expensive
 - Optimise the whole, efficient flow of requirements into products
 - Respect people, support people and build expertise

Summary

- Respecting People
 - Focus on developing skills
- Create knowledge
 - Learning plays an important role
- Build quality in
 - Commitment to high quality

Summary

- Value
 - Identify value in development process
- Waste
 - Remove work items that don't add value
- Speed
 - Quickly deliver value to clients

Summary

- Lean Start-up
 - Bootstrap, Focus on revenue to build tech start-ups
 - Use mock-ups to assess customer reaction
 - Minimum viable product, essence of solution
 - Pivot, change direction if needed