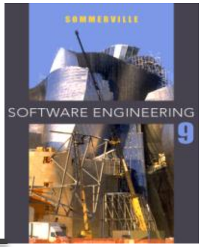
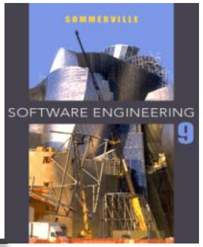


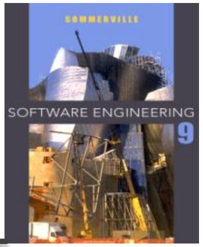
Chapter 3 – Agile Software Development

Part 2

Chapter 3 – Agile Software Development



Topics covered

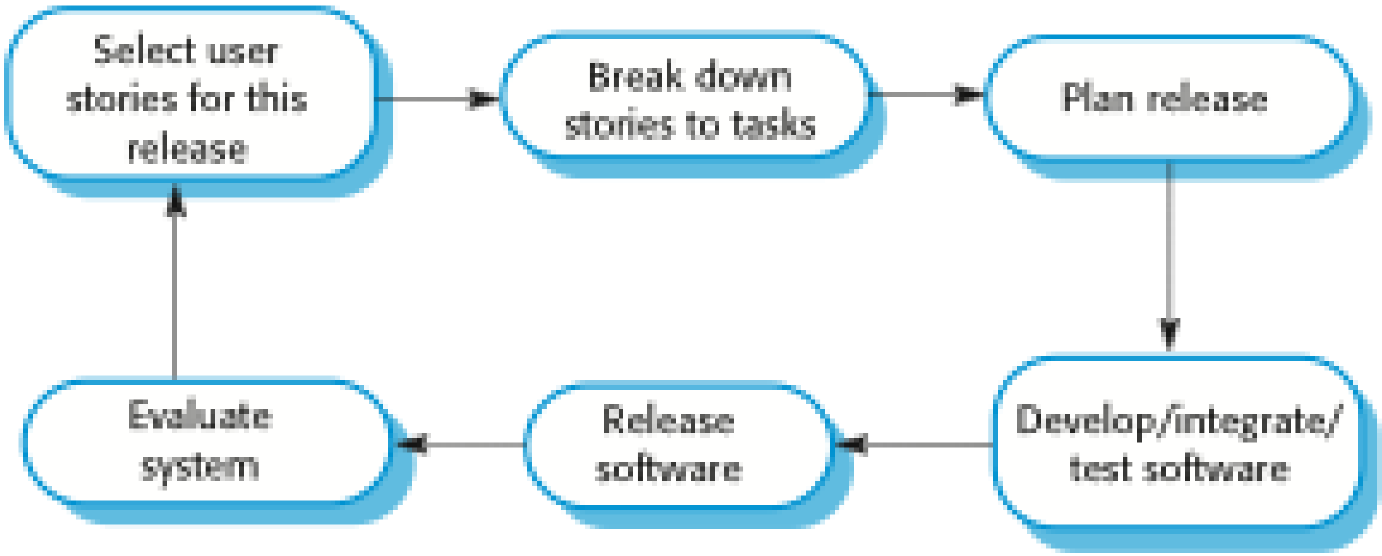


- Intro
- 3.1 Agile methods
- 3.2 Plan-driven and agile development
- 3.3 Extreme programming
- 3.4 Agile project management
- 3.5 Scaling agile methods

3.3 Extreme programming

- Perhaps the best-known and most widely used agile method.
- 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 successfully executed when new code is integrated into the system.

The extreme programming release cycle (Fig 3.3)


Select user stories for this release
Break down stories to tasks
Plan release
Develop/integrate/ test software
Release software
Evaluate system

Extreme programming practices

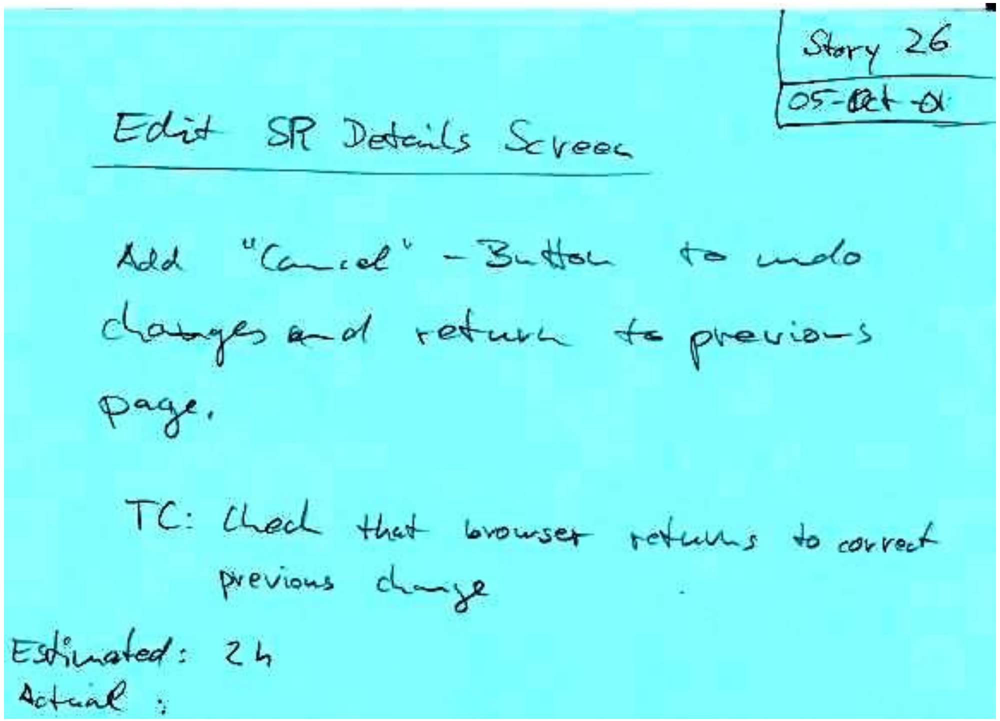
Incremental planning

TODO: explain upcoming exercise

Incremental planning

Extreme programming practices (Fig 3.4)

Incremental planning



Story 26

05-Oct-01

Edit SR Details Screen

Add "Cancel"-Button to undo changes and return to previous page.

TC: Check that browser returns to correct previous change

Estimated: 2h
Actual :

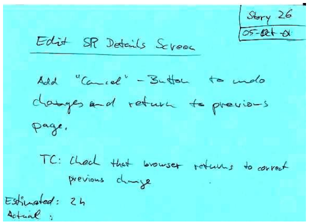
User Stories on cards

Basis of schedule and cost estimate

user chooses which stories to implement next

Extreme programming practices (Fig 3.4)

Incremental planning



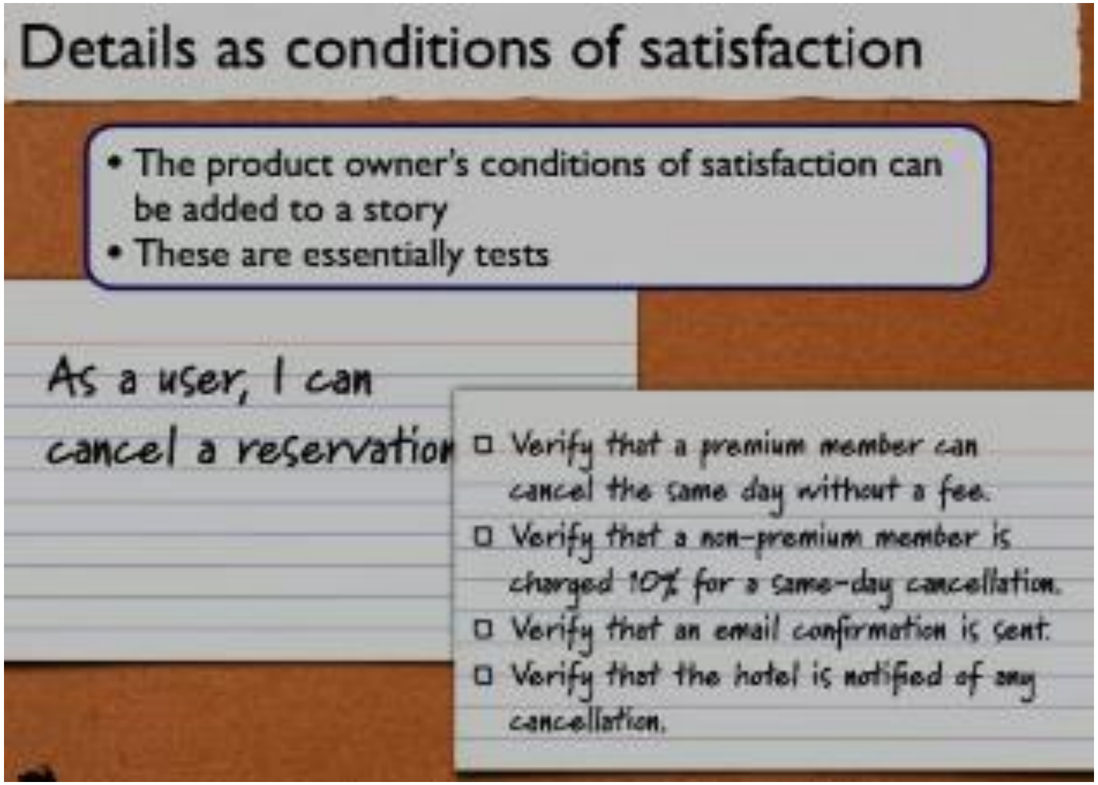
Story 26
05-Oct-01

Edit SR Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that browser returns to correct previous change

Estimated: 2h
Actual :



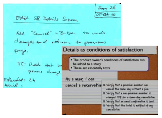
Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

□ Verify that a premium member can cancel the same day without a fee.
□ Verify that a non-premium member is charged 10% for a same-day cancellation.
□ Verify that an email confirmation is sent.
□ Verify that the hotel is notified of any cancellation.

Extreme programming practices (Fig 3.4)



Story 26

05-Oct-0X

Edit SR Details Screen

Add "Cancel"-Button to undo changes and return to previous page.

TC: Check that w

previous change

Estimated: 2.4

Actual:

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

□ Verify that a premium member can cancel the same day without a fee.

□ Verify that a non-premium member is charged 10% for a same-day cancellation.

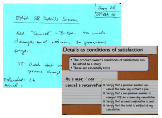
□ Verify that an email confirmation is sent.

□ Verify that the hotel is notified of any cancellation.

Incremental planning

Small releases

Extreme programming practices (Fig 3.4)



Story 26
05-Oct-0x

Edit SR Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that [illegible] previous change[illegible]

Estimated: 2 h
Actual :

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

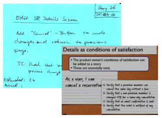
□ Verify that a premium member can cancel the same day without a fee.
□ Verify that a non-premium member is charged 10% for a same-day cancellation.
□ Verify that an email confirmation is sent.
□ Verify that the hotel is notified of any cancellation.

Incremental planning

days or weeks

Small releases

Extreme programming practices (Fig 3.4)



Story 26
05-Okt-0x

Edit SR Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that w
previous change

Estimated: 24
Actual :

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

□ Verify that a premium member can cancel the same day without a fee.
□ Verify that a non-premium member is charged 10% for a same-day cancellation.
□ Verify that an email confirmation is sent.
□ Verify that the hotel is notified of any cancellation.

Incremental planning

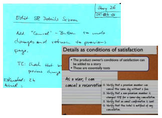
User involvement

days or weeks

Small releases

Test-first development

Extreme programming practices (Fig 3.4)



Incremental planning

days or weeks

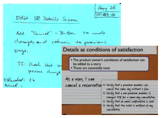
Small releases

User Involve-ment

Test-first development

Refactoring

Extreme programming practices (Fig 3.4)



Incremental planning

days or weeks

User Involve-ment

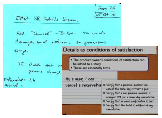
Test-first d

Simple design

Everything should be made as simple as possible, but no simpler.”

attributed to Albert Einstein

Extreme programming practices (Fig 3.4)



Story 26

05-Oct-0X

Edit SR Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that ... previous change ...

Estimated: 2 h

Actual :

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

□ Verify that a premium member can cancel the same day without a fee.
□ Verify that a non-premium member is charged 10% for a same-day cancellation.
□ Verify that an email confirmation is sent.
□ Verify that the hotel is notified of any cancellation.

Incremental planning

days or weeks

Small releases

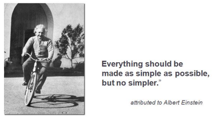
User Involve-ment

Test-first development

On-site customer

Refactoring

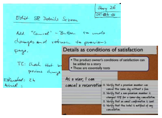
Simple design



Everything should be made as simple as possible, but no simpler."

attributed to Albert Einstein

Extreme programming practices (Fig 3.4)



Edit SR Details Screen

Story 26

05-Oct-0[illegible]

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that [illegible] previous change[illegible]

Estimated: 24

Actual :

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

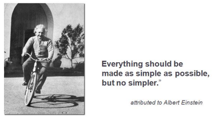
- □ Verify that a premium member can cancel the same day without a fee.
- □ Verify that a non-premium member is charged 10% for a same-day cancellation.
- □ Verify that an email confirmation is sent.
- □ Verify that the hotel is notified of any cancellation.

Incremental planning

User Involve-ment

Test-first development

Simple design

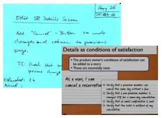


Everything should be made as simple as possible, but no simpler."

attributed to Albert Einstein

Pair programming

Extreme programming practices (Fig 3.4)



Story 26
05-Oct-0x

Edit SR Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that w... previous change...

Estimated: 24
Actual :

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

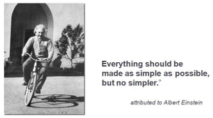
- □ Verify that a premium member can cancel the same day without a fee.
- □ Verify that a non-premium member is charged 10% for a same-day cancellation.
- □ Verify that an email confirmation is sent.
- □ Verify that the hotel is notified of any cancellation.

Incremental planning

User Involve-ment

Test-first development

Simple design

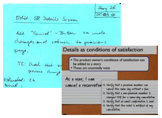


Everything should be made as simple as possible, but no simpler."

attributed to Albert Einstein

Pair programming

Extreme programming practices (Fig 3.4)



Story 26
05-Oct-0[illegible]

Edit SR Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that [illegible] previous change [illegible]

Estimated: 2 h
Actual:

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

□ Verify that a premium member can cancel the same day without a fee.
□ Verify that a non-premium member is charged 10% for a same-day cancellation.
□ Verify that an email confirmation is sent.
□ Verify that the hotel is notified of any cancellation.

Incremental planning

days or weeks

Small releases

User Involve-ment

Test-first development

On-site customer

Refactoring

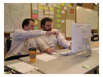
Simple design

Everything should be made as simple as possible, but no simpler."

attributed to Albert Einstein

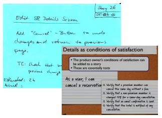
Collective ownership

Pair programming



Extreme programming practices (Fig 3.4)

Extreme Programming



Story 26
05-Oct-[illegible]

Edit SR Details Screen

Add "Cancel"-Button to undo changes and return to previous page.

TC: Check that w
previous change

Estimated: 24
Actual :

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservatio

- □ Verify that a premium member can cancel the same day without a fee.
- □ Verify that a non-premium member is charged 10% for a same-day cancellation.
- □ Verify that an email confirmation is sent.
- □ Verify that the hotel is notified of any cancellation.

Incremental planning

Sustainable pace

days or weeks

Small releases

User Involve-ment

Test-first development

On-site customer

Refactoring

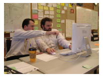
Simple design

Everything should be made as simple as possible, but no simpler."

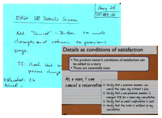
attributed to Albert Einstein

Collective ownership

Pair programming



Extreme programming practices (Fig 3.4)



Story 26

05-Oct-0X

Edit SR Details Screen

Add "Cancel" - Button to undo changes and return to previous page.

TC: Check that w
previous change

Estimated: 2h
Actual :

Details as conditions of satisfaction

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

As a user, I can cancel a reservation

□ Verify that a premium member can cancel the same day without a fee.
□ Verify that a non-premium member is charged 10% for a same-day cancellation.
□ Verify that an email confirmation is sent.
□ Verify that the hotel is notified of any cancellation.

Incremental planning

Continuous integration

Sustainable pace

days or weeks

Small releases

User Involve-ment

Test-first development

On-site customer

Refactoring

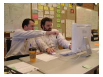
Simple design

Everything should be made as simple as possible, but no simpler.”

attributed to Albert Einstein

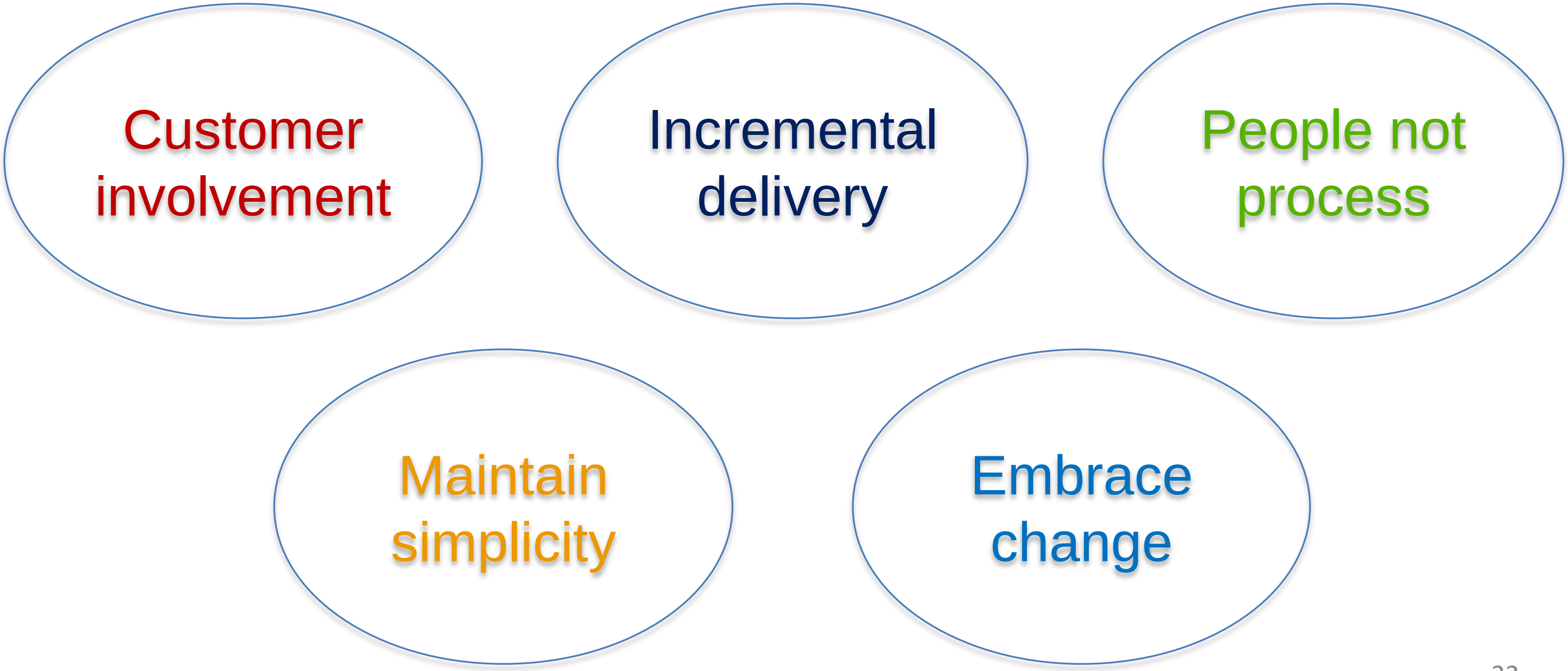
Collective ownership

Pair programming



The principles of agile methods

✧ All agile methods (XP, Scrum, Crystal, . . .) share principles based on the agile manifesto



Customer involvement

Incremental delivery

People not process

Maintain simplicity

Embrace change

XP and change

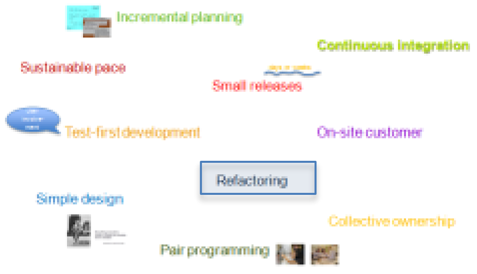
✧ Conventional wisdom: design for change.

✧ XP: changes cannot be reliably anticipated

➡ simple design

➡ refactor

Refactoring



- Improves quality of code without changing functionality
- Improves understandability
  => reduces need 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.

3.3.1 Testing in XP

✧ XP testing features:

- **Test-first** development.
- Incremental test development from scenarios.
- User involvement in test development and validation.
- Automated test harnesses used
- Acceptance testing (with user data) incremental

3.3.1 Testing in XP

Incremental planning
Continuous integration
Sustainable pace
Small releases
Test-first development
On-site customer
Refactoring
Simple design
Collective ownership
Pair programming

✧ Difficulties:

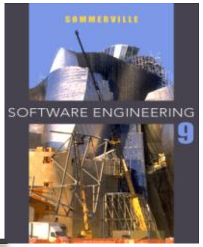
- Programmers prefer programming to testing
- Some tests are difficult to write incrementally
- Completeness

3.3.2 Pair programming

- ✧ programmers work in pairs, sitting side by side to develop code. (pairs created dynamically)
- ✧ collective ownership of code spreads knowledge across the team.
- ✧ informal code review
- ✧ encourages refactoring

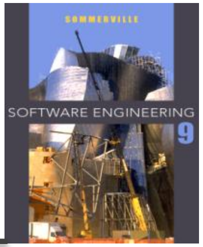
3.4 Agile project management

3.4 Agile project management



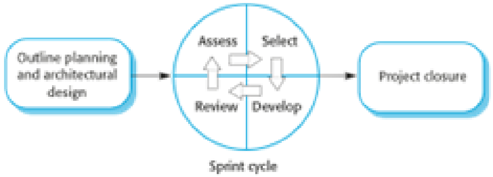
- Principal responsibility of software project managers:
  - Deliver on time within budget
  - Monitor progress
  - Supervise developers
- Standard approach: plan-driven.

3.4 Agile project management

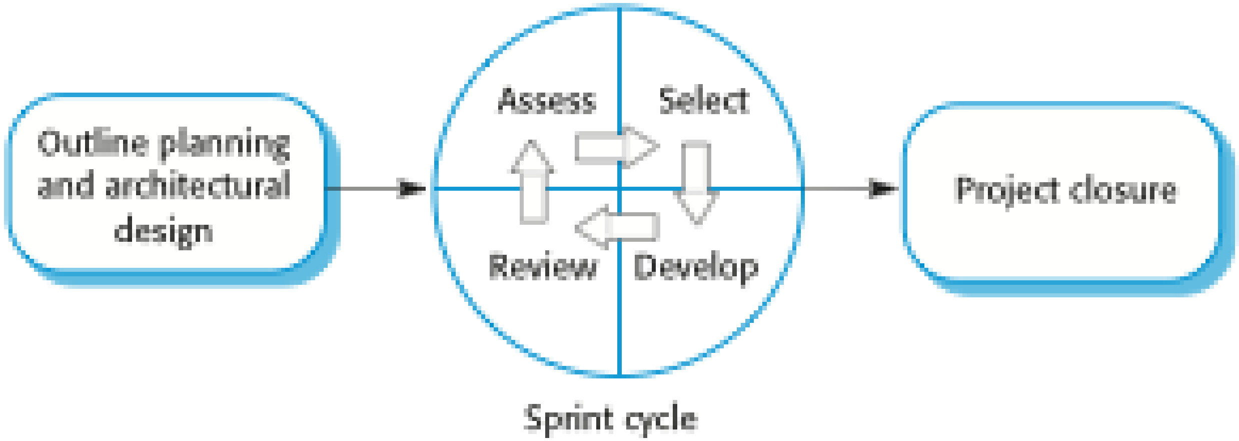


- ✧ Principal responsibility of software project managers:
  - Deliver on time within budget
  - Monitor progress
  - Supervise developers
- ✧ Standard approach: plan-driven.
- ✧ Agile project management: requires different approach

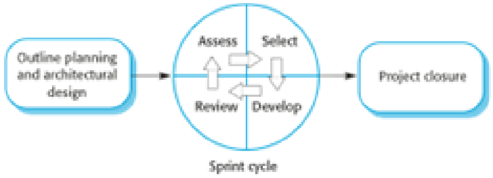
3.4 Scrum



- ✧ **Scrum** approach is a general agile method
- ✧ No prescribed programming practices like pair programming
- ✧ Provides management framework for iterative development
- ✧ Sprint .. Planning unit of fixed length; (few weeks)

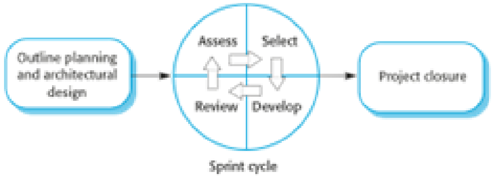


3.4 Sprint



- ✧ Assess: review product backlog, priorities, risks
- ✧ Select: features and functionality to be developed
- ✧ Develop: team organizes themselves
  short daily stand-up meeting
  Scrum master protects from external distractions
- ✧ Review: work reviewed and presented to stakeholders

3.4 Sprint



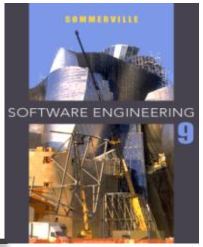
✧ Idea: whole team empowered to make decisions

✧ Scrum master is facilitator

- Arranges meetings, tracks work, records decisions, ..

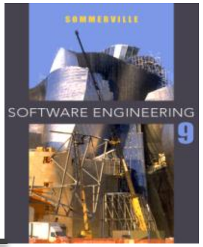
✧ Daily meetings: (short, often stand-up)

- What was accomplished yesterday
- What will be done today
- Any problem that hold me back



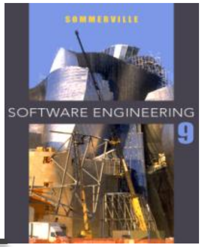
SCRUM VIDEO: (2 min)

http://www.youtube.com/watch?v=WxiuE-1ujCM&feature=related



3.5 Scaling agile methods

3.5 Scaling agile methods



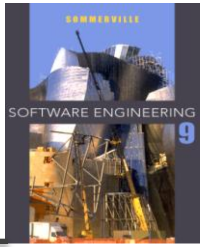
- **Scaling up**

  to large software systems

- **Scaling out**

  across large organization

3.5 Scaling agile methods



- **Scaling up** (to large software systems)

  Distributed / diverse stakeholders / multiple systems /

  continuous integration impractical / rules regulations

- **Scaling out** (across large organization)

  reluctant to accept risk / company culture / diverse skills

  quality procedures / mandated tools