3.

Agenda

- **ISTQB – 7 principles of testing**
- Software development lifecycle (SDLC) models
- Software testing lifecycle (STLC)
- Specification based basic test design techniques

1. Testing shows presence of defects

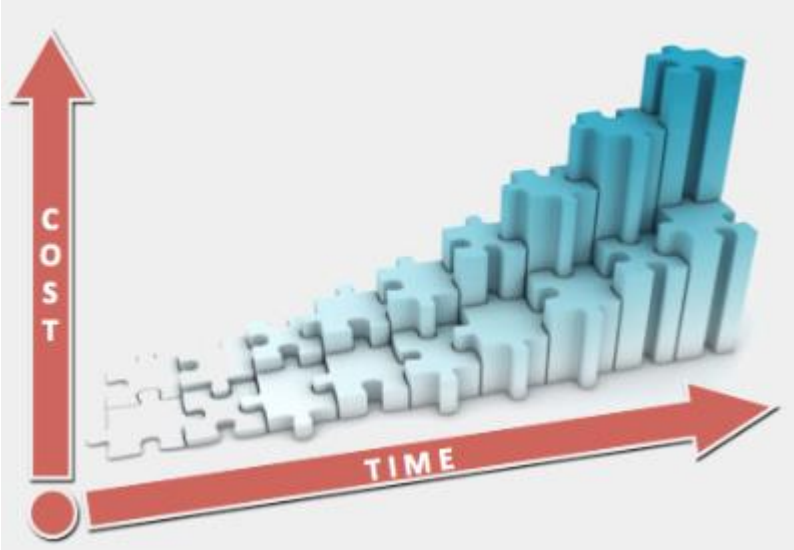
- Testing can show that defects are present, BUT CANNOT PROVE that there is NO DEFECTS
- Testing REDUCES THE PROBABILITY OF undiscovered defects remaining in the software
- BUT, even if no defect are found, it is not a proof of correctness

2. Exhaustive testing is impossible

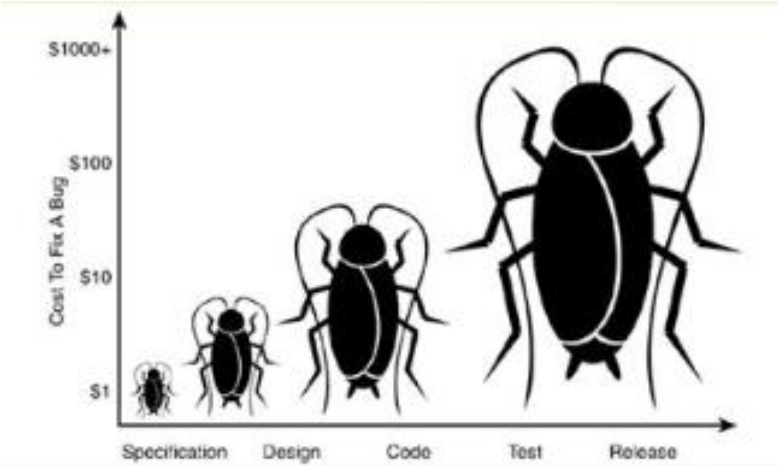
- Testing everything (all combinations of inputs and preconditions) is NOT FEASIBLE except for trivial cases.

- Instead of exhaustive testing, RISK ANALYSIS AND PRIORITIES should be used to focus testing efforts.

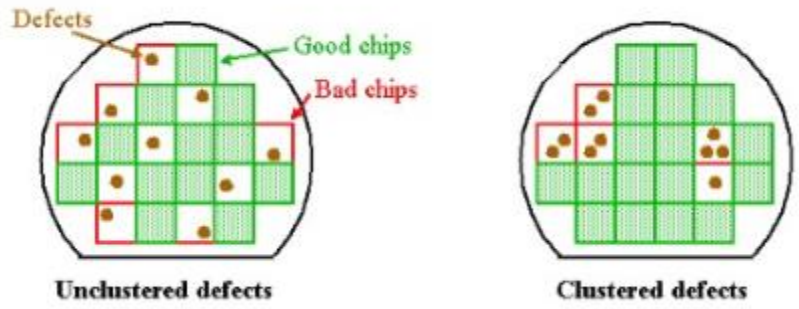
3. Early testing



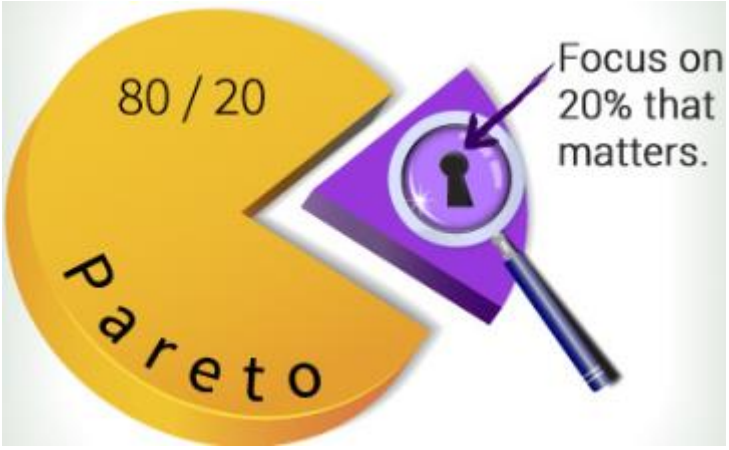
- To find defects early, testing activities shall be started AS EARLY AS POSSIBLE in the software or system development life cycle, and shall be focused on defined objectives.



4. Defect clustering



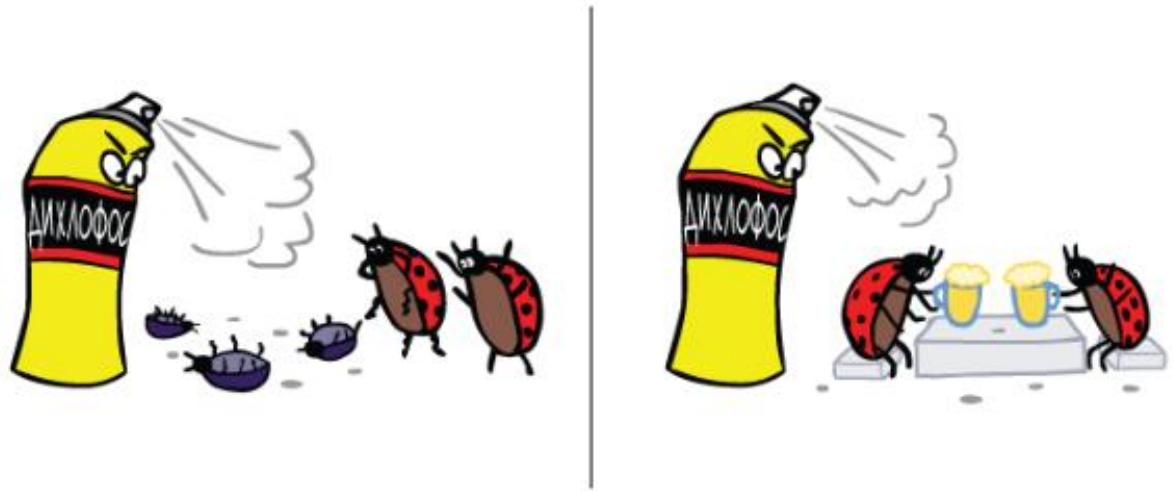
- Defects are not evenly distributed in the system – they are 'clustered'



- A small number of modules usually contains most of the defects discovered during testing, or shows the most of operational failures.

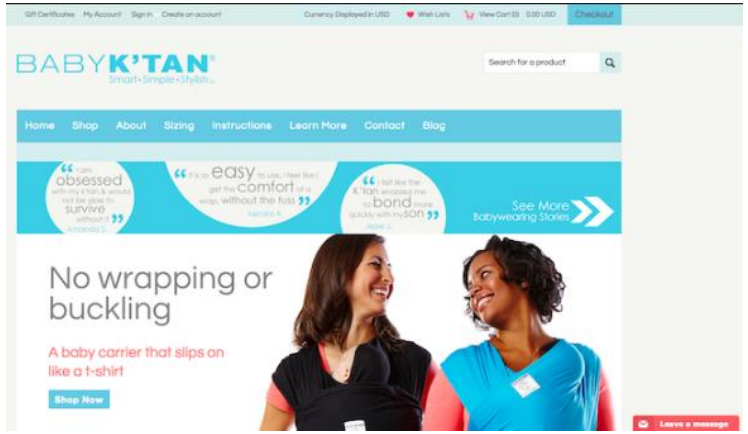
5. Pesticide paradox

- If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects.
- To overcome this test cases need to be regularly reviewed and revised.
- New and different tests need to be written to exercise different parts of the software or system to find potentially more defects.



6. Testing is context dependent

- Testing is done differently in different contexts.
- For example, safety-critical software is tested differently from an e-commerce site.



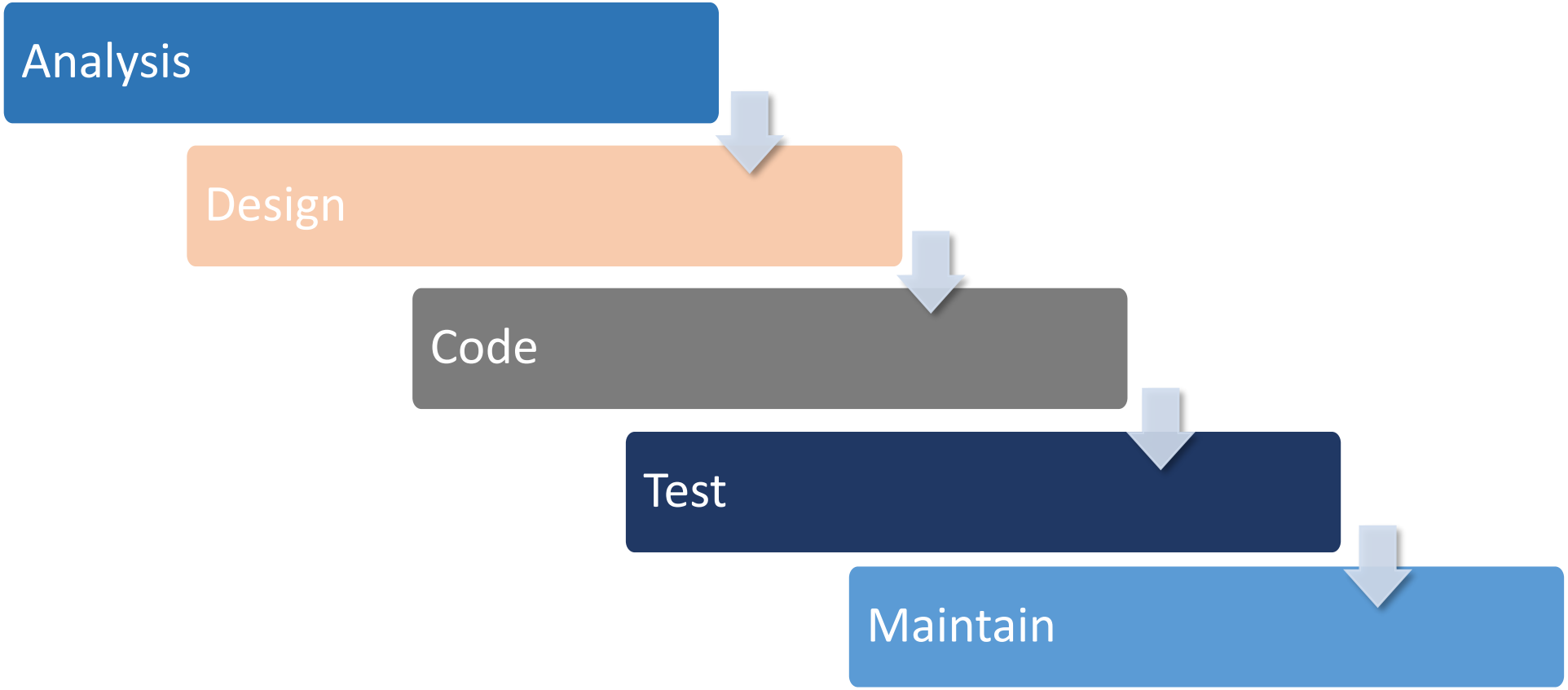
7. Absence-of-errors fallacy

- Finding and fixing defects does not help if the system built is unusable and does not fulfill the users needs and expectations.

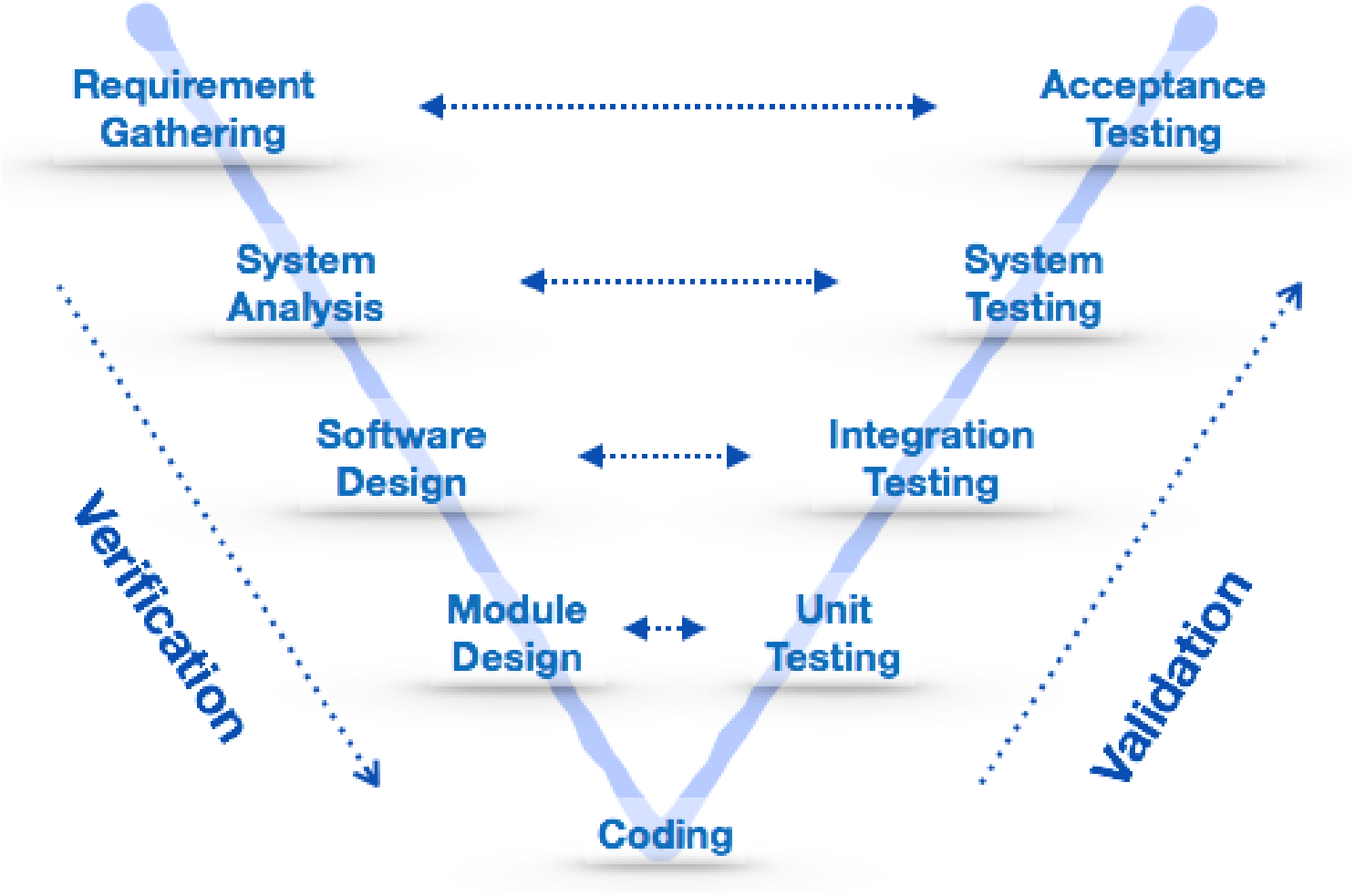
Agenda

- ISTQB – 7 principles of testing
- **Software development lifecycle (SDLC) models**
- Software testing lifecycle (STLC)
- Specification based basic test design techniques

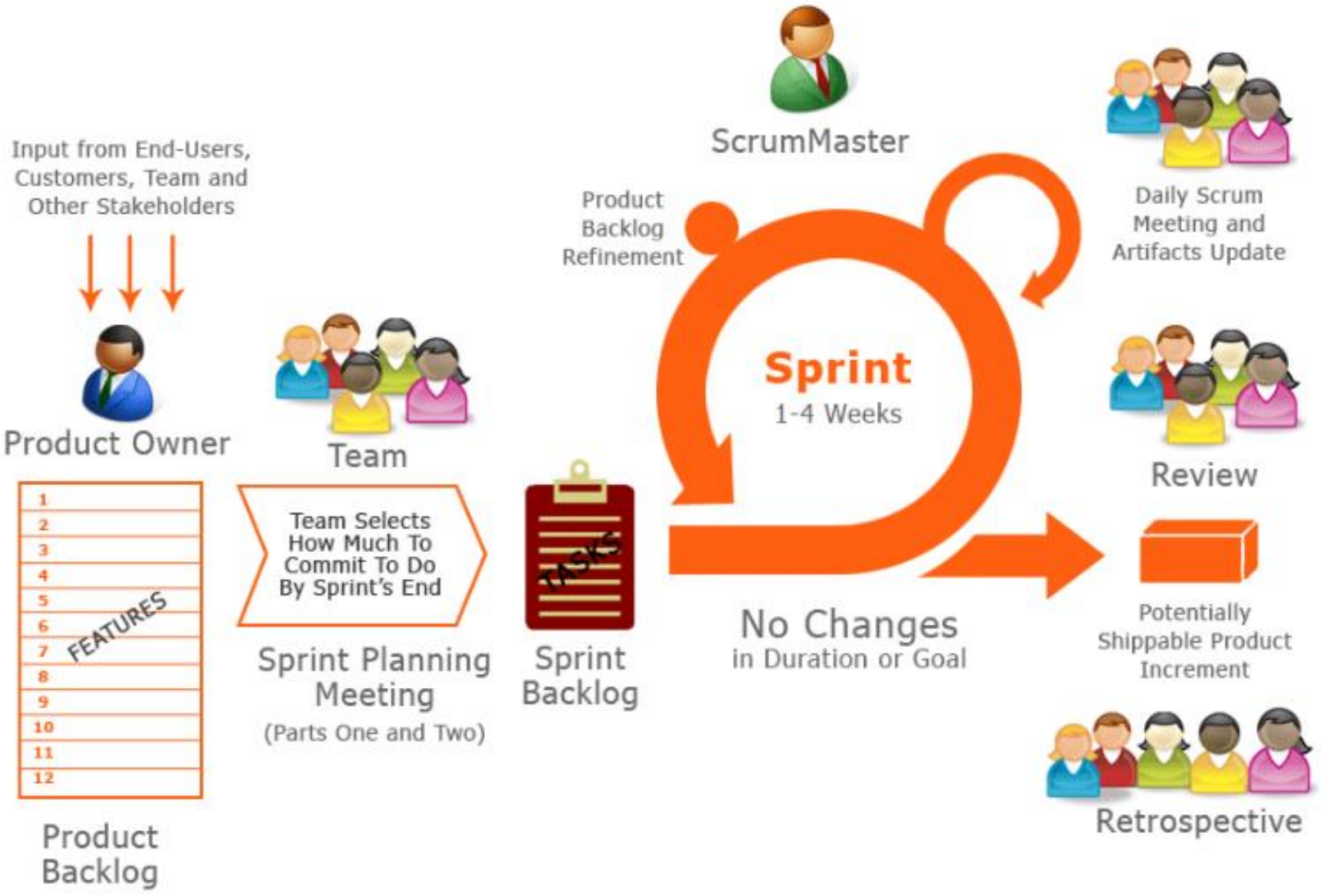
SDLC – Waterfall model



SDLC – V model


Requirement Gathering
Acceptance Testing
System Analysis
System Testing
Software Design
Integration Testing
Module Design
Unit Testing
Coding
Verification
Validation

SDLC – Agile: Scrum model



JIRA Software

Teams in Space
Scrum: Teams in Space

- Backlog
- Agile board
- Releases
- Reports
- All issues
- Components
- Add-ons

PROJECT SHORTCUTS

- Mars Team HipChat Room
- Space Station Dev Roadmap
- Teams in Space Org Chart
- Orbital Spotify Playlist
- Hyperspeed Bitbucket Repo
- + Add shortcut

Backlog

Configure

QUICK FILTERS: Product Recently updated Only my issues Server UI

VERSIONS

EPICS

All issues

- SeeSpaceEZ Plus
- Large Team Support
- Space Travel Partners
- Summer Saturn Sale
- Afterburner Plus
- Local Mars Office
- Hyper-speed shuttles
- New launch platforms
- Delicious Space Nutrition
- Spacetainment

Sprint 1 14 issues 3 6 5

Sprint 2 6 issues Start sprint

Start: 10 Aug 2015 — Release: 9 Oct 2015

TIS-25	Engage Jupiter Express for outer solar system travel	SeeSpaceEZ Plus	5
TIS-37	When requesting user details the service should return prior trip info	Large Team Support	1
TIS-9	After 100,000 requests the SeeSpaceEZ server dies	Local Mars Office	1
TIS-7	500 Error when requesting a reservation	Large Team Support	1
TIS-10	Bad JSON data coming back from hotel API	Space Travel Partners	5
TIS-18	Enable Speedy SpaceCraft as the preferred individual transit provider	Large Team Support	1

Backlog 49 issues Create sprint

TIS-25	Engage Jupiter Express for outer solar system travel	Local Mars Office	5
TIS-37	When requesting user details the service should return prior trip info	Space Travel Partners	1
TIS-9	After 100,000 requests the SeeSpaceEZ server dies	Space Travel Partners	1
TIS-7	500 Error when requesting a reservation	Local Mars Office	1

Scrum Board

7 days remaining Complete Sprint

QUICK FILTERS: Product UI Server Only My Issues Recently Updated

2 To Do

TIS-68
Homepage footer uses an inline style - should use a class
Large Team Support 4

TIS-16
Establish relationship with local office supplies company
Local Mars Office 9

3 In Progress

TIS-49
Draft network plan for Mars Office
Local Mars Office 5

TIS-17
Engage Saturn's Rings Resort as a preferred provider
Space Travel Partners 7

TIS-30
Create Saturn Summer Sizzle Logo
Summer Saturn Sale 1

TIS-23
Engage JetShuttle SpaceWays for short distance space travel
Local Mars Office 1

2 Code Review

TIS-69
Add a string anonymizer to TextUtils
Large Team Support 1

TIS-67
Developer Toolbox does not display by default
Large Team Support 5

1 Done

~~TIS-8~~
Requesting available flights is now taking > 5 seconds
Large Team Support 2

~~TIS-56~~
Add pointer to main css file to instruct user to create child themes
Large Team Support 4

~~TIS-45~~
Email non registered users to sign up with Teams in Space
Summer Saturn Sale 8

QUICK FILTERS: Only My Issues Recently Updated

To Do

XAI-5
As a user, I can reset my password
NOK 5

XAI-30
Sub Test Execution for XAI-5
ios

XAI-31
Sub Test Execution for XAI-5
android

XAI-8
As a user, I can access my JIRA projects from the app
UNCOVERED 2

XAI-12
As a user, I can send text messagem to other users
UNCOVERED 5

Implementing

XAI-6
As a user, I can export JIRA issues to PDF
UNCOVERED 10

XAI-11
As a user, I can search for JIRA issues using JQL
UNCOVERED 5

Waiting for Testing

XAI-1
As a user, I can login into the app using the fingerprint scanner
NOTRUN 31

XAI-22
Sub Test Execution for XAI-1
android

XAI-2
As a user, I can login into the app using Facebook authentication
NOTRUN 10

XAI-7
As a system, I must provide a mechanism anti bot
NOTRUN 10

XAI-38
Sub Test Execution for XAI-7
ios

XAI-39
Sub Test Execution for XAI-7
android

Testing

XAI-1 As a user, I can login into the app using the fingerprint s...

XAI-21
Sub Test Execution for XAI-1
ios

XAI-2 As a user, I can login into the app using Facebook auth...

XAI-16
Sub Test Execution for XAI-2
android

Done

XAI-2 As a user, I can login into the app using Facebook auth...

~~XAI-15~~
Sub Test Execution for XAI-2
ios

~~XAI-3~~
As a user, I can login into the app using Google+ authentication
OK 5

~~XAI-27~~
Sub Test Execution for XAI-3
ios

~~XAI-28~~
Sub Test Execution for XAI-3
android

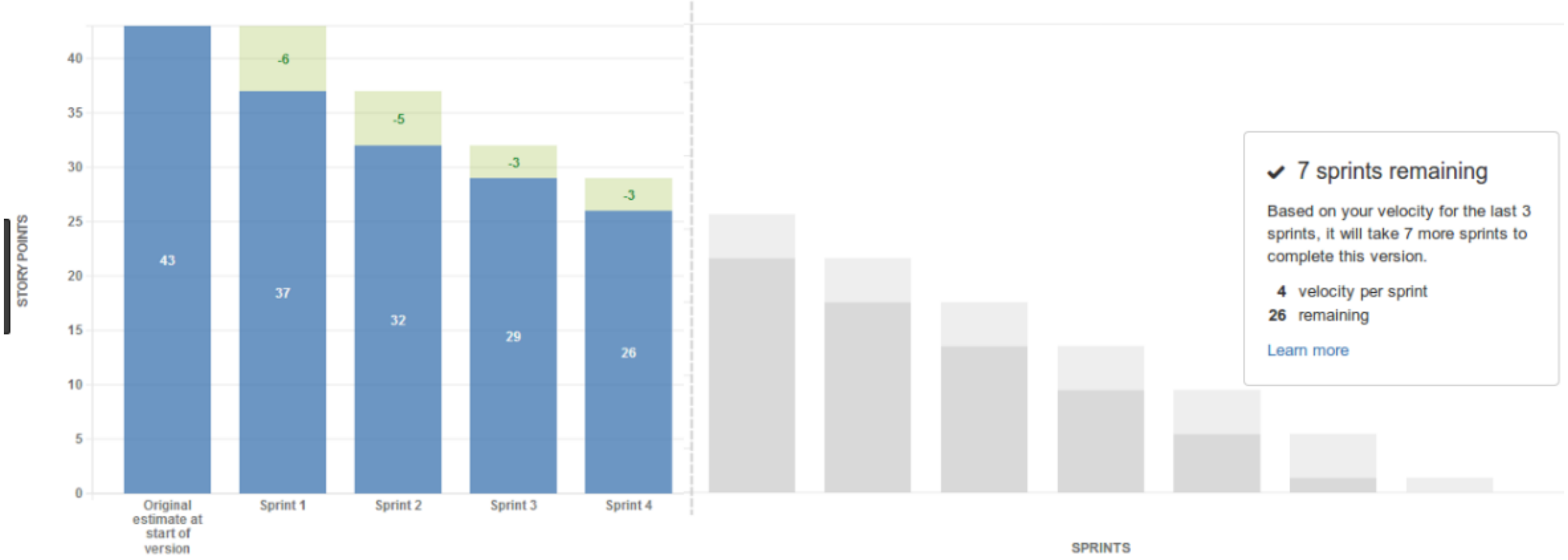
~~XAI-4~~
As an admin, I can give special permissions to users
OK 5

~~XAI-24~~
Sub Test Execution for XAI-4
ios

~~XAI-25~~
Sub Test Execution for XAI-4
android

Release Burndown SSP: Version 1.0 How to read this chart

0% unestimated issues **22** remaining (story points)



Kanban Board

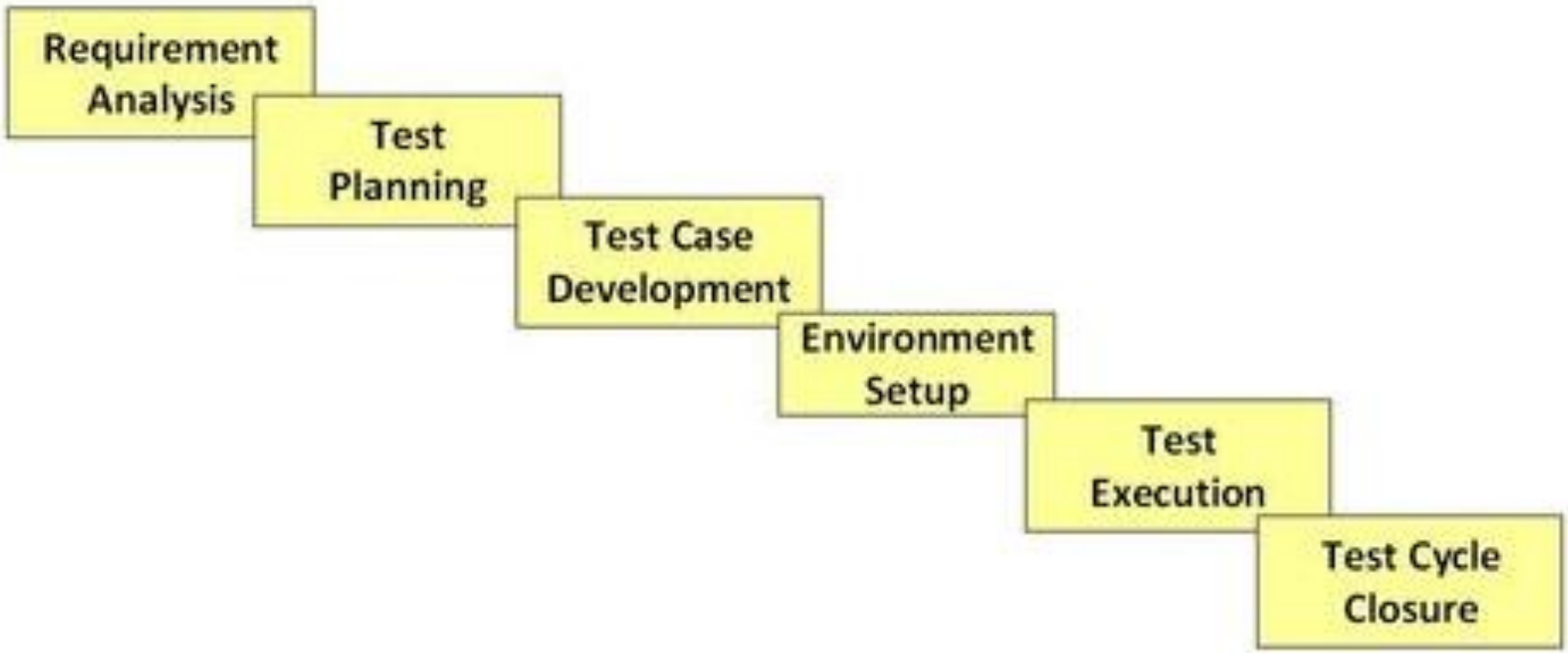
112 To Do	**5** In Progress	**4** Done
Expedite		
TIS-12 Create 90 day plans for all departments in the Mars Office Local Mars Office	TIS-9 After 100,000 requests the SeeSpaceEZ server dies SeeSpaceEZ Plus	TIS-72 Add video chat interface
Everything Else		
~~TIS-8~~ Requesting available flights is now taking > 5 seconds SeeSpaceEZ Plus	TIS-49 Draft network plan for Mars Office Local Mars Office	~~TIS-63~~ Book now button is disabled
TIS-20 Engage Saturn Shuttle Lines for group tours Space Travel Partners	TIS-68 Homepage footer uses an inline style - should use a class Local Mars Office	
TIS-45 Email non registered users to sign up with Teams in Space Space Travel Partners		

Agenda

- ISTQB – 7 principles of testing
- Software development lifecycle (SDLC) models
- **Software testing lifecycle (STLC)**
- Specification based basic test design techniques

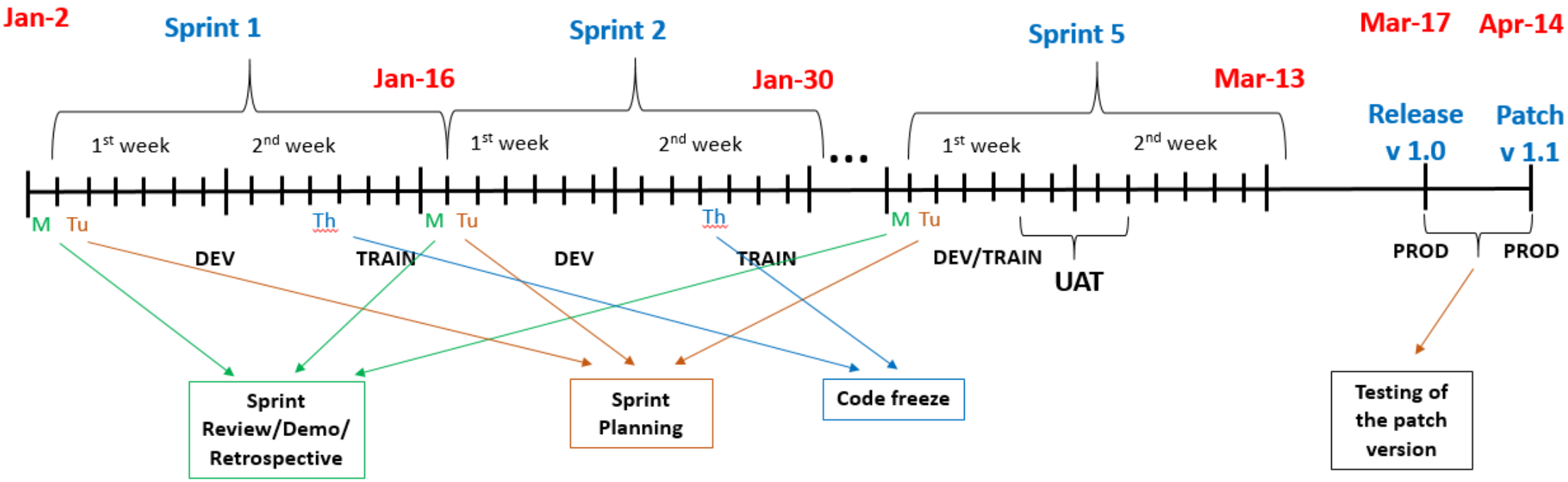
Software testing lifecycle (STLC)

STLC - sequence of activities conducted to perform Software Testing.



SCRUM? SCRUM?
SCRUM? SCRUM? SCRUM?
SCRUM?

“Calculator” Development example



Sprint 1 – implementation of “+” functionality
Sprint 2 – implementation of “-” functionality
Sprint 3 – implementation of “*” functionality
Sprint 4 – implementation of “/” functionality
Sprint 5 – stabilization sprint (mostly bug fixing)

QA:
1st week – specification review/impact analysis, test cases preparation, automation, execution once a story/defect fix is ready (acceptance/sanity/re-testing), daily stand-up meetings
2nd week – test cases preparation, automation, execution once a story/defect fix is ready (acceptance/sanity/re-testing), regression testing after the code freeze, daily stand-up meetings
Sprint 5 – regression testing and re-testing of fixed bugs

Agenda

- ISTQB – 7 principles of testing
- Software development lifecycle (SDLC) models
- Software testing lifecycle (STLC)
- **Specification based basic test design techniques**

OH NOOOO
NOT AGAIN

Reminder: Test design techniques

Static

testing of a component or system at specification or implementation level **without execution** of that software, e.g. reviews or static code analysis

Dynamic

testing that involves the **execution** of the software of a component or system

structure-based

specification-based

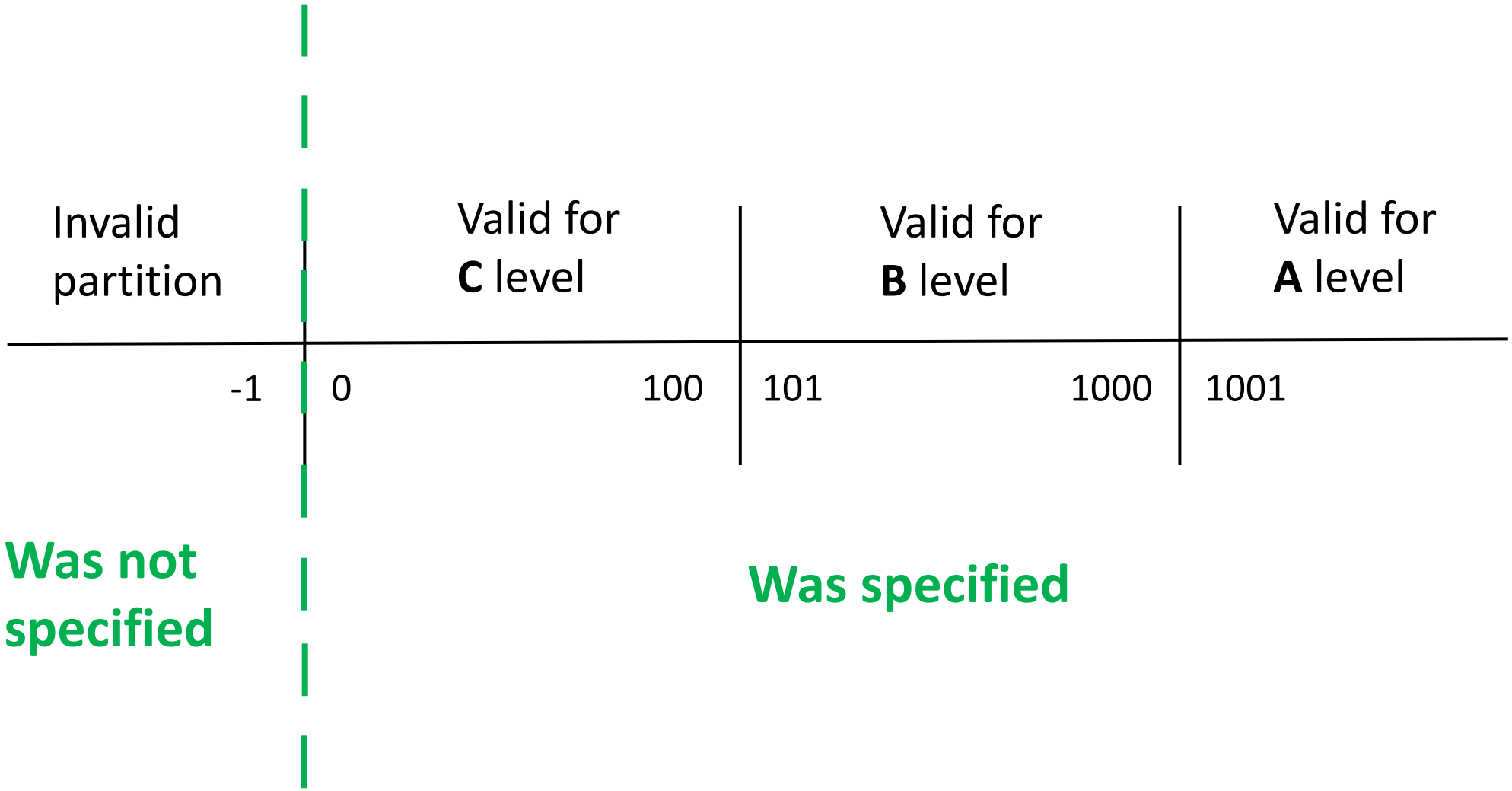
experience-based

Let’s design some tests

Exam evaluation score corresponds to following levels:

-If a score is in the range 0 to 100 – **C level**
-If a score is in the range 101 to 1000 – **B level**
-If a score over 1000 – **A level**

Equivalence partitions (classes)



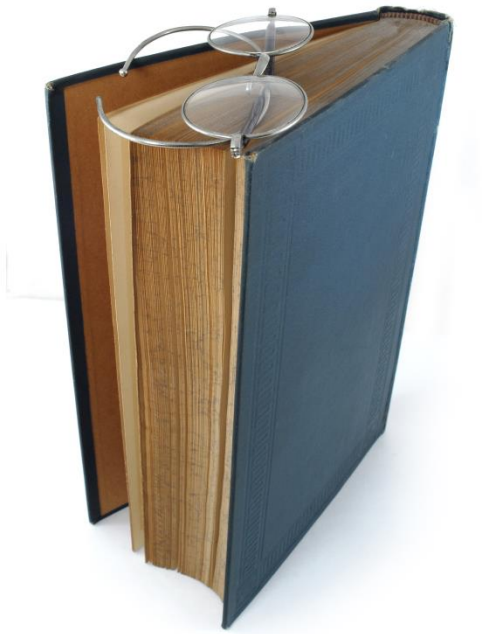
Equivalence partitions (classes)

Invalid partition	Valid for **C** level	Valid for **B** level	Valid for **A** level
-1	0 100	101 1000	1001

At least one value from each partition:

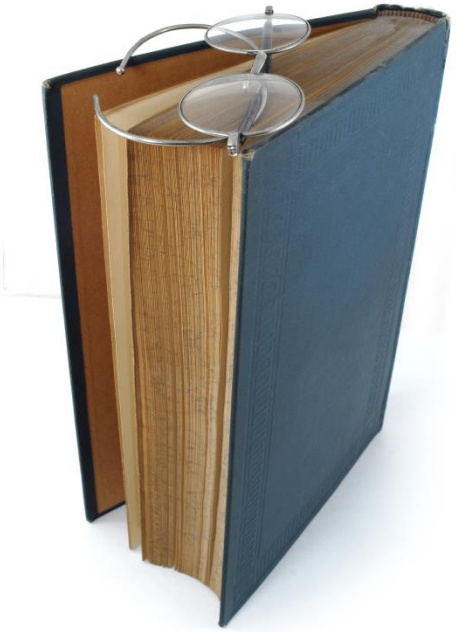
-10 ; 50 ; 486; 1389

Equivalence partitioning



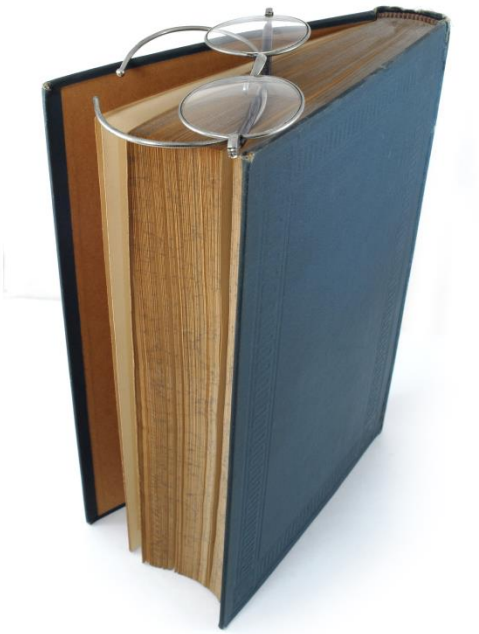
Equivalence partitioning divides data into partitions (also known as equivalence classes) in such a way that all the members of a given partition are expected to be processed in the same way.

Equivalence partitioning



With this technique test cases must cover all identified partitions (including invalid partitions) by using a minimum of one value from each partition.

Equivalence partitions (classes)

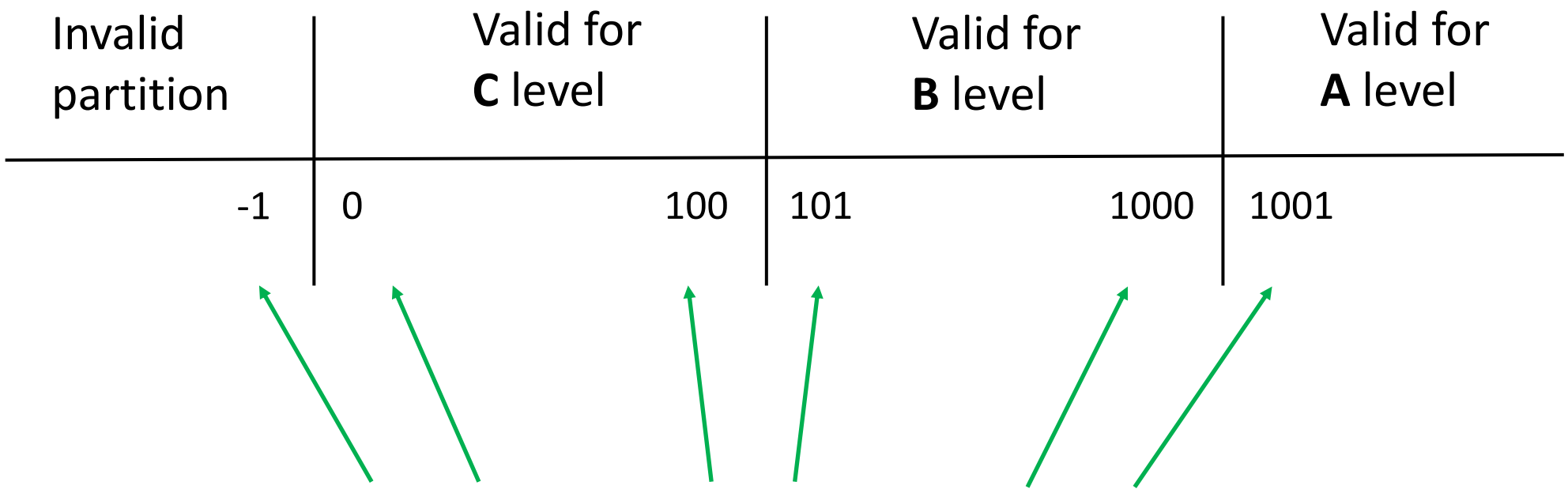


A portion of an input or output domain for which the behavior of a component of system is assumed to be the same, based on the specification.

Best representatives

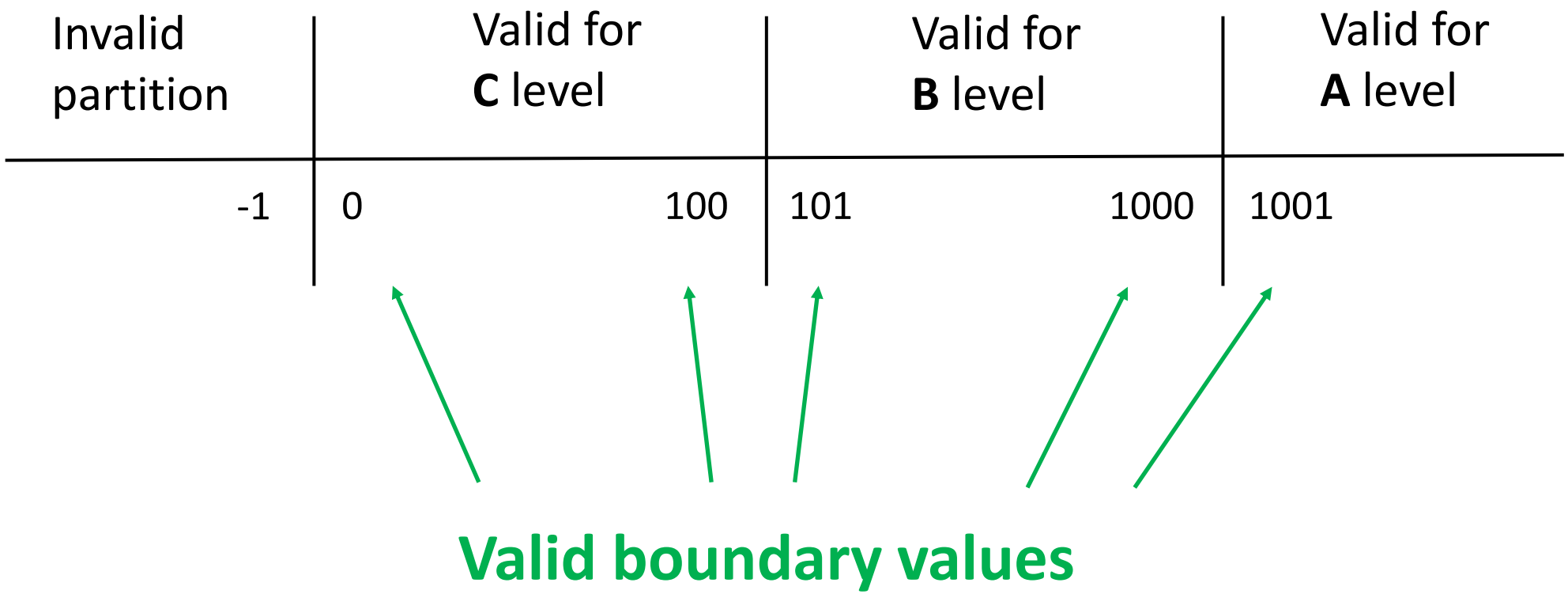
If we only test one value from each partition, which is the best one?

Equivalence partitions (classes)



These are boundary values

Equivalence partitions (classes)



Equivalence partitions (classes)

Invalid partition	Valid for **C** level	Valid for **B** level	Valid for **A** level
-1	0 100	101 1000	1001

Invalid boundary value

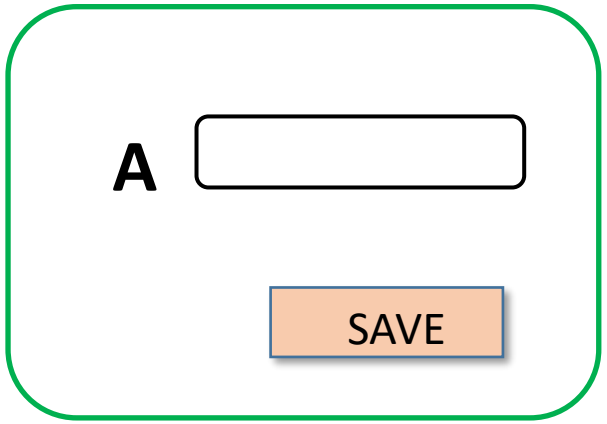
Equivalence partitions (classes)

Invalid partition	Valid for **C** level	Valid for **B** level	Valid for **A** level
-1	0 100	101 1000	1001

Open boundaries

We need to find them too!

Homework Exercise 1



On [SAVE] button click:

IF A > 3000
THEN display error message “A is too big!”
ELSE
    IF A < 0
    THEN display error message “A cannot be negative!”
    ELSE successful save

TO DO:

1. Identify equivalences partitions (classes)
2. Identify boundary values for each class

Let’s design some tests

A savings account in a bank earns a different rate of interest:

-If a balance in the range $0.00 to $99.99 – **3%**
-If a balance in the range $100.00 to $1000.00 – **5%**
-If a balance over $1000.00 – **7%**

What if…

If value is $0.07,

expected interest

is $0.00

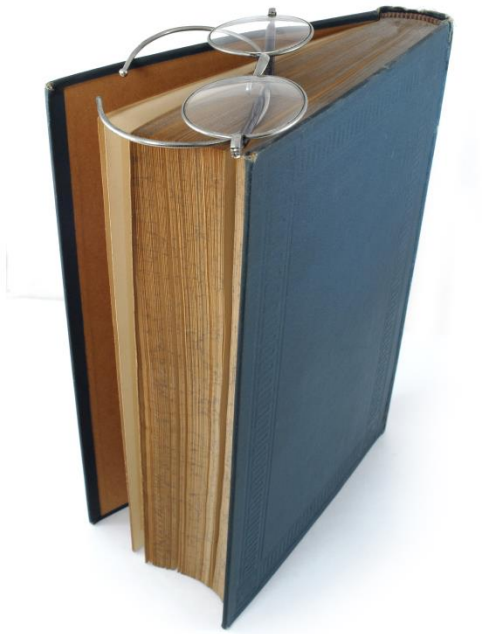
regardless of 3% or 7%

Output equivalence analysis

	Class 1`	Class 1``
Input	$0.00 … $0.16	$0.17 … $99.99
Output	$0.00	$0.01 … $3.00

Are these boundary values too?

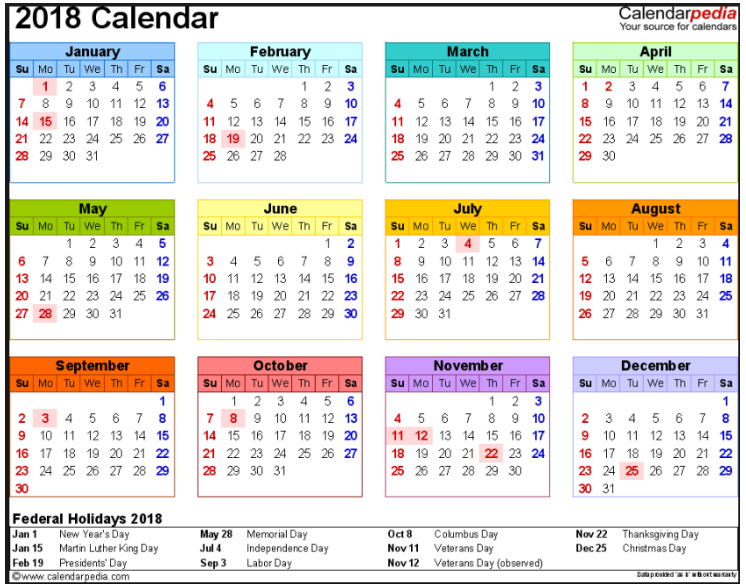
Boundary value analysis



Boundary value analysis (BVA) is an extension of equivalence partitioning, but can only be used when the partition is ordered.

The minimum and maximum values of a partition are its boundary values.

Homework Exercise 2



Date field of the format:

DD/MM/YYYY

TO DO:

Test the date field