

Testing Course Agenda

- Learn basic concepts to understand testing process and work
- Requirement engineering and management
- Agile methods and testing
- Test design & execution
- Test Planning & Management
- Test Automation
- Continuous Integration & Testing
- Understand how and why testing is important

- This material if for general training for Test Design and management
- Material is more supportive in class room
- Material will be updated during courses
- FreeNest Portable Project Platform is used to demonstrate things only in practice.
This is not limiting usage for material for other training environments (I hope ☺)

ISTQB

- <http://www.rbcus-us.com/images/documents/The-ISTQB-Advanced-Syllabus.pdf>

Verification & Validation

Validation = Are we building the right product?

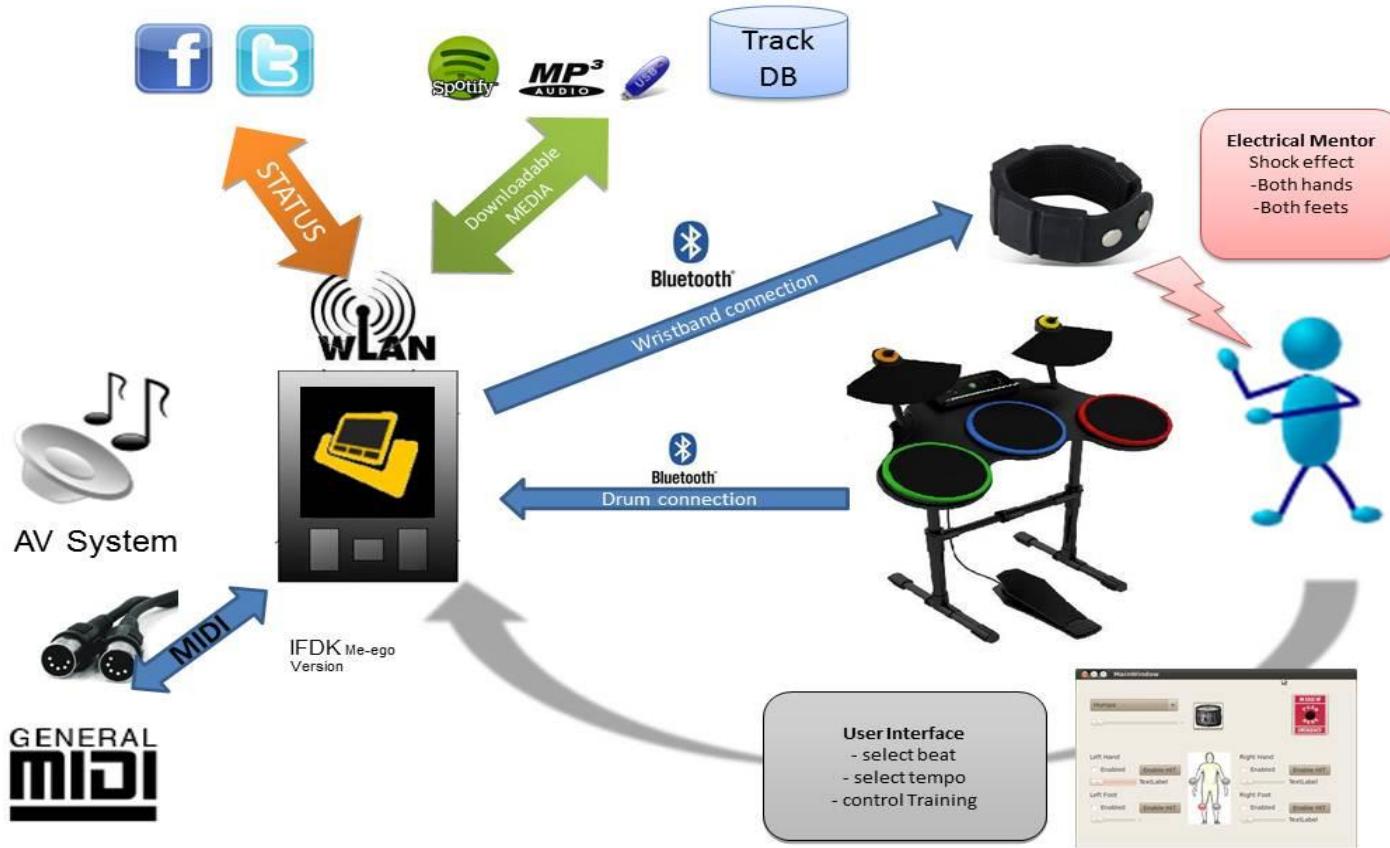
Verification = Are we building the product right?

Requirement Engineering

- We take branding seriously, The FreeNest Brand is our pride and joy.
- It is the message we want to send to the developer teams, customers and stakeholders.
- These guidelines must be followed in all FreeNest's print and electronic communications, marketing and informing.
- Documentation defines logo guidelines, typography and the use of colors.

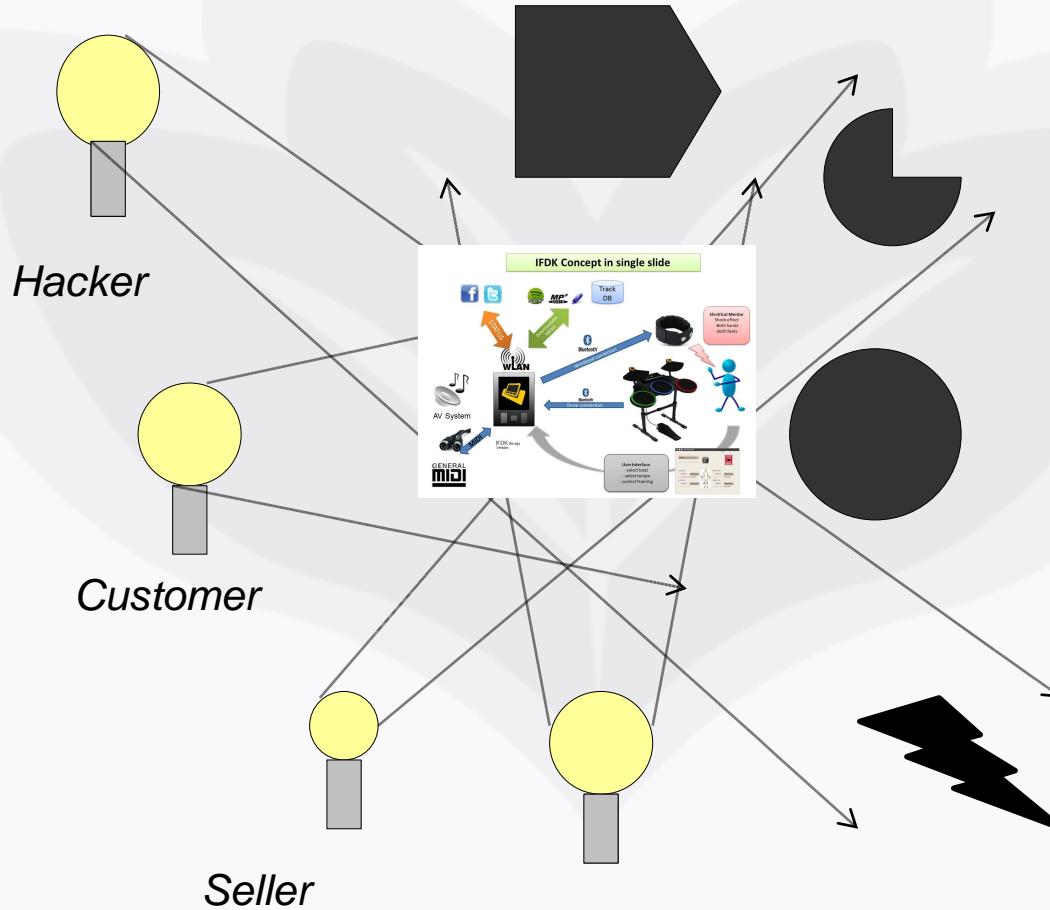
System Requirements?

IFDK Concept in single slide



FreeNest

Different points of view on the product

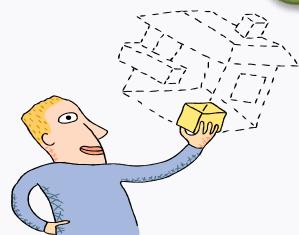


Different levels of requirements

Customer Business Stakeholder



System



Design

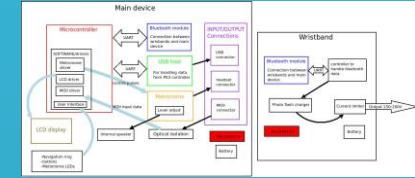
Components



Implementation

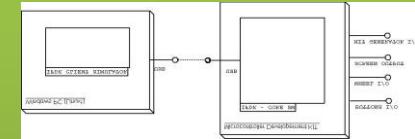
Requirements in different levels

System Engineering



SysML

Software Engineering - Architecture

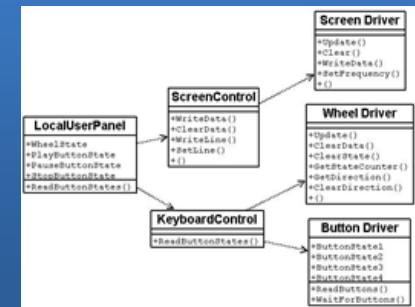


UML

Software Engineering - Design

Code

.NET, JAVA,
C++



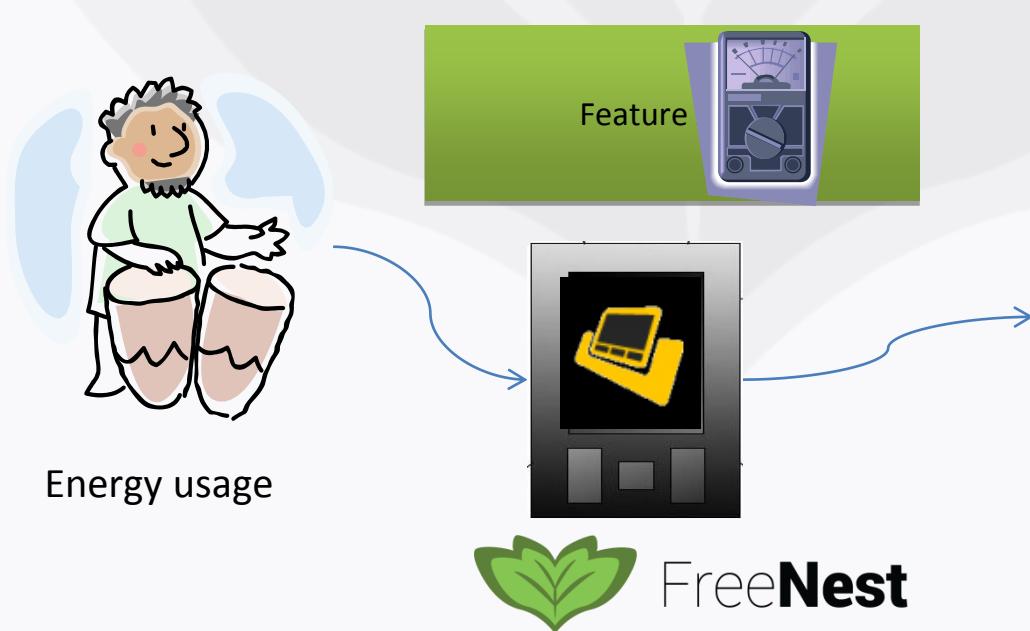
<http://www.sysml.org/>

Google: SysML, UML, Systems engineering, Design,

What is feature?

Calory Counter:

Player can measure calories during training session. This can be seen as exercise result in web service eg. Facebook application



Is product a combination of features?

Table Drum Mode

Standby Mode

MIDI Support

Calory Counter

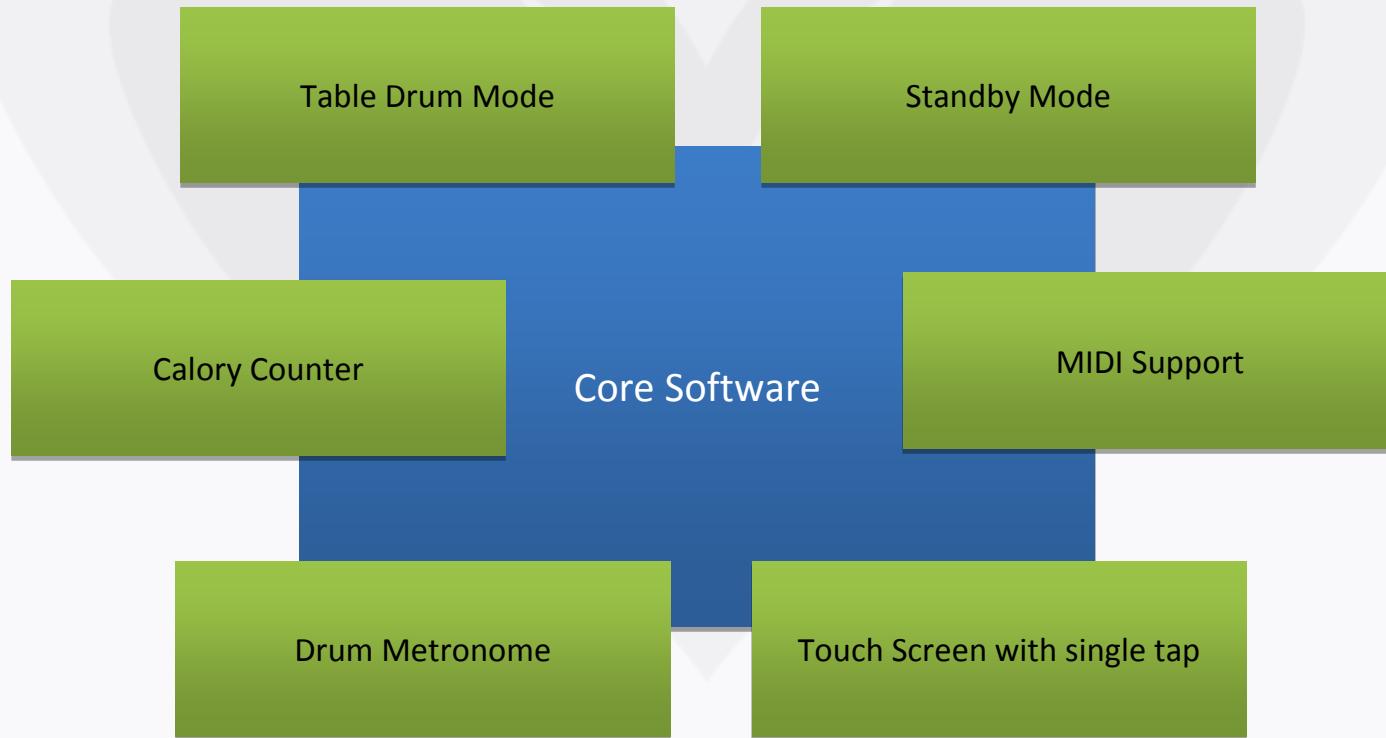
Drum Metronome

Touch Screen with single tap



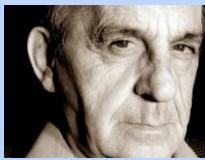
Free**Nest**

Is product a combination of features



Free**Nest**

Customer/Business Requirements



Touch Screen with single tap

Simple Training Mode

Drum Metronome

Calory Counter

MIDI Support

Table Drum Mode

Customer Type 1

Customer Type 2

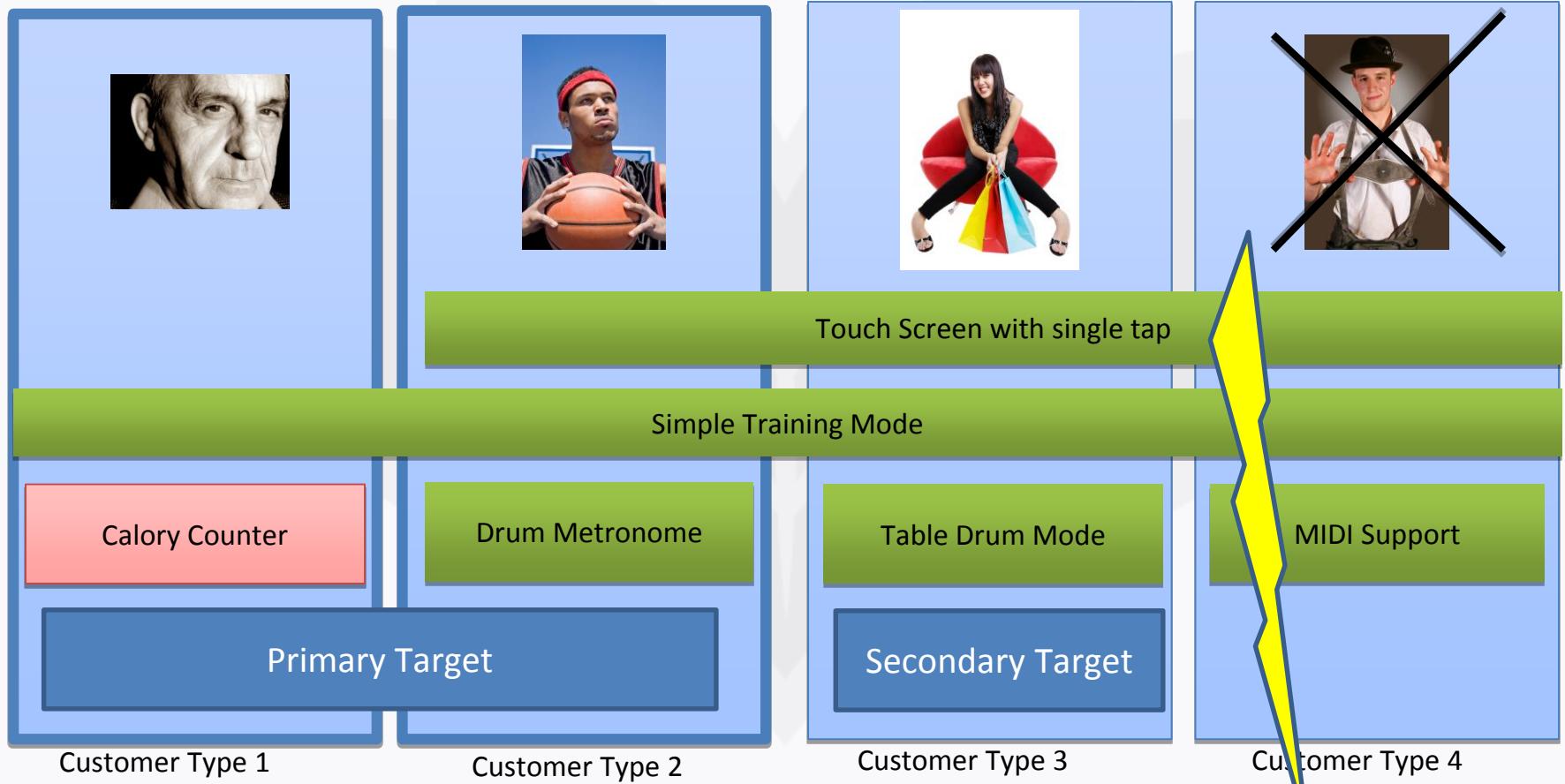
Customer Type 3

Customer Type 4



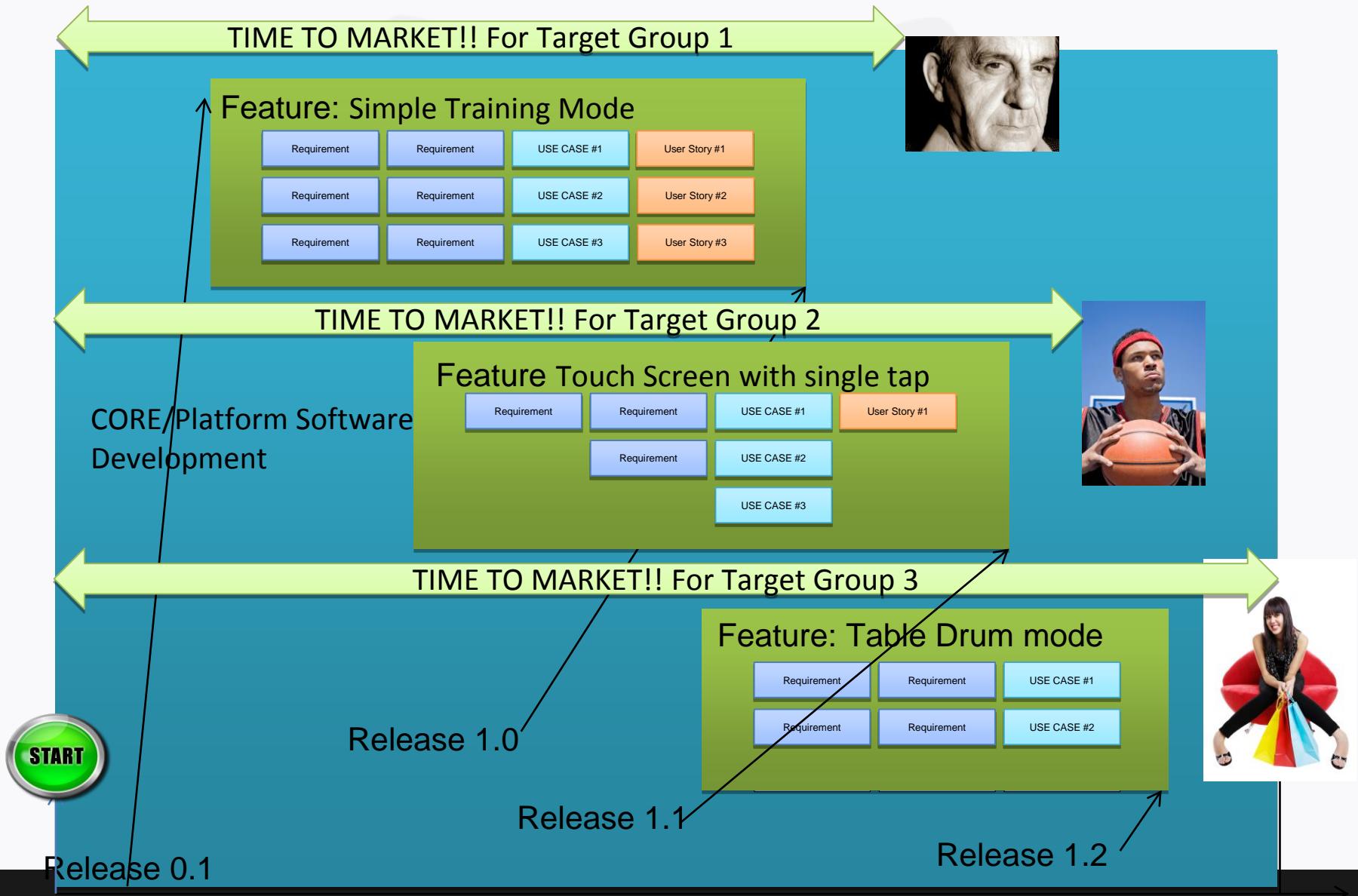
Quest Who are our target customers?

Customer strategy

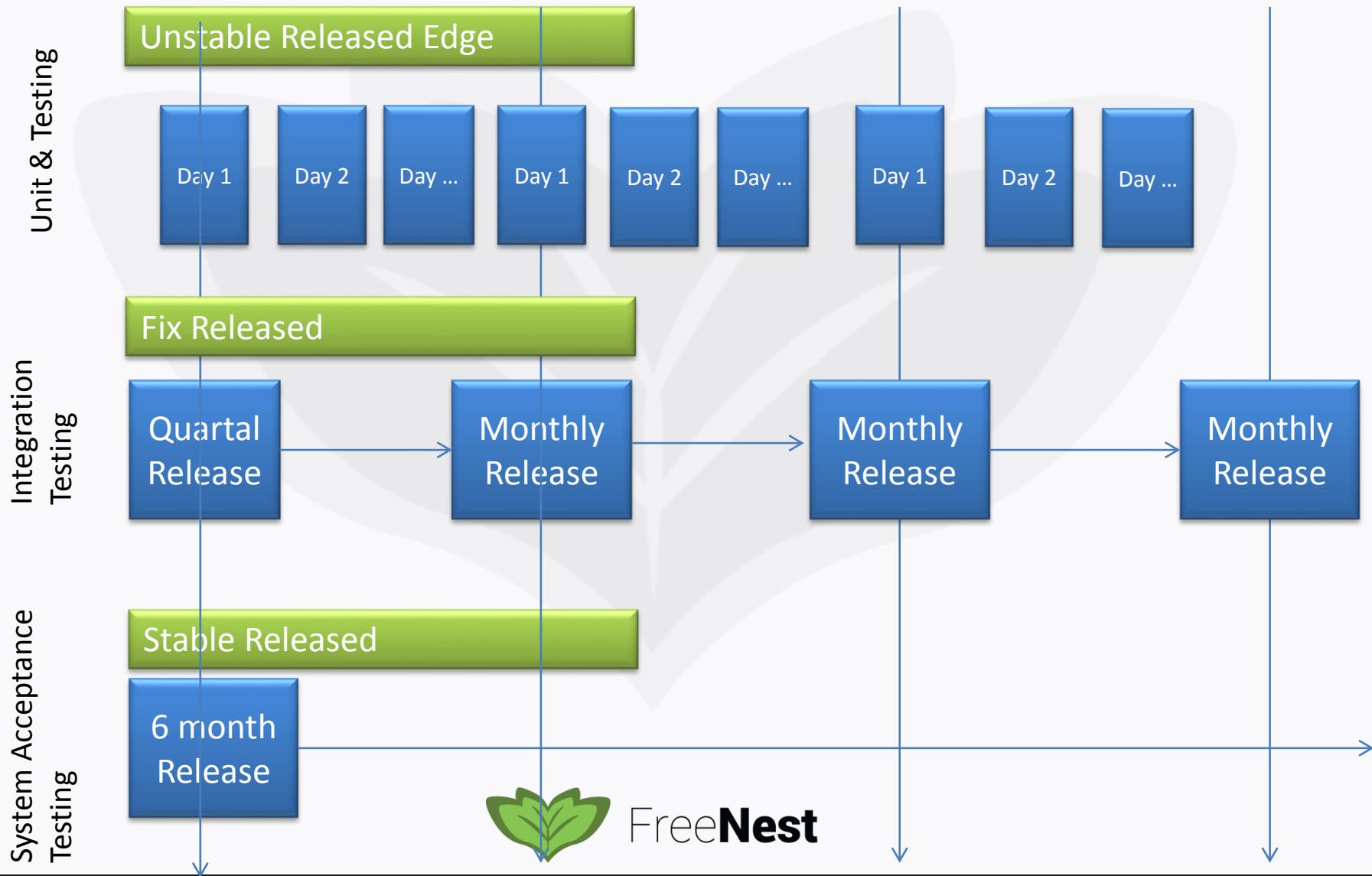


Free Nest
What is our key customer?

Features and roadmap



Release plan for large system



SW Project vs Open Source

Open Source – Crowd Sourcing

SW Release tested without coordination by group of volunteers

Release tested by customer

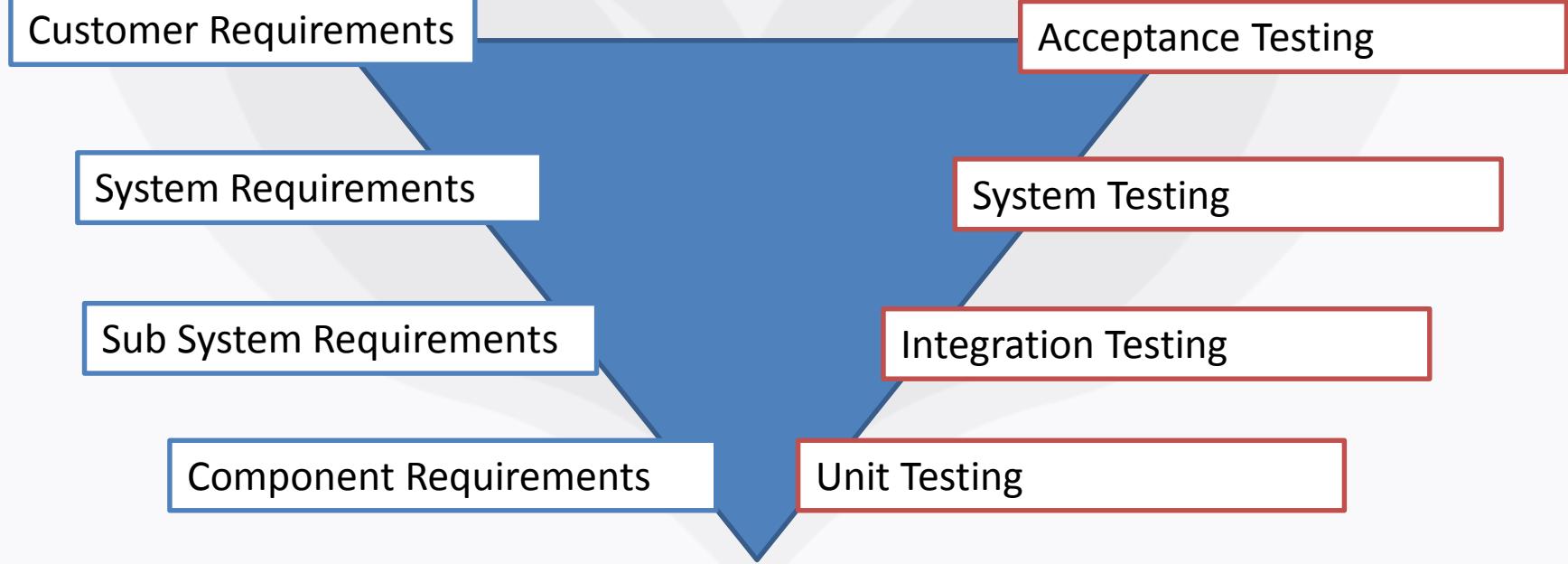
Field Testing

Test Group

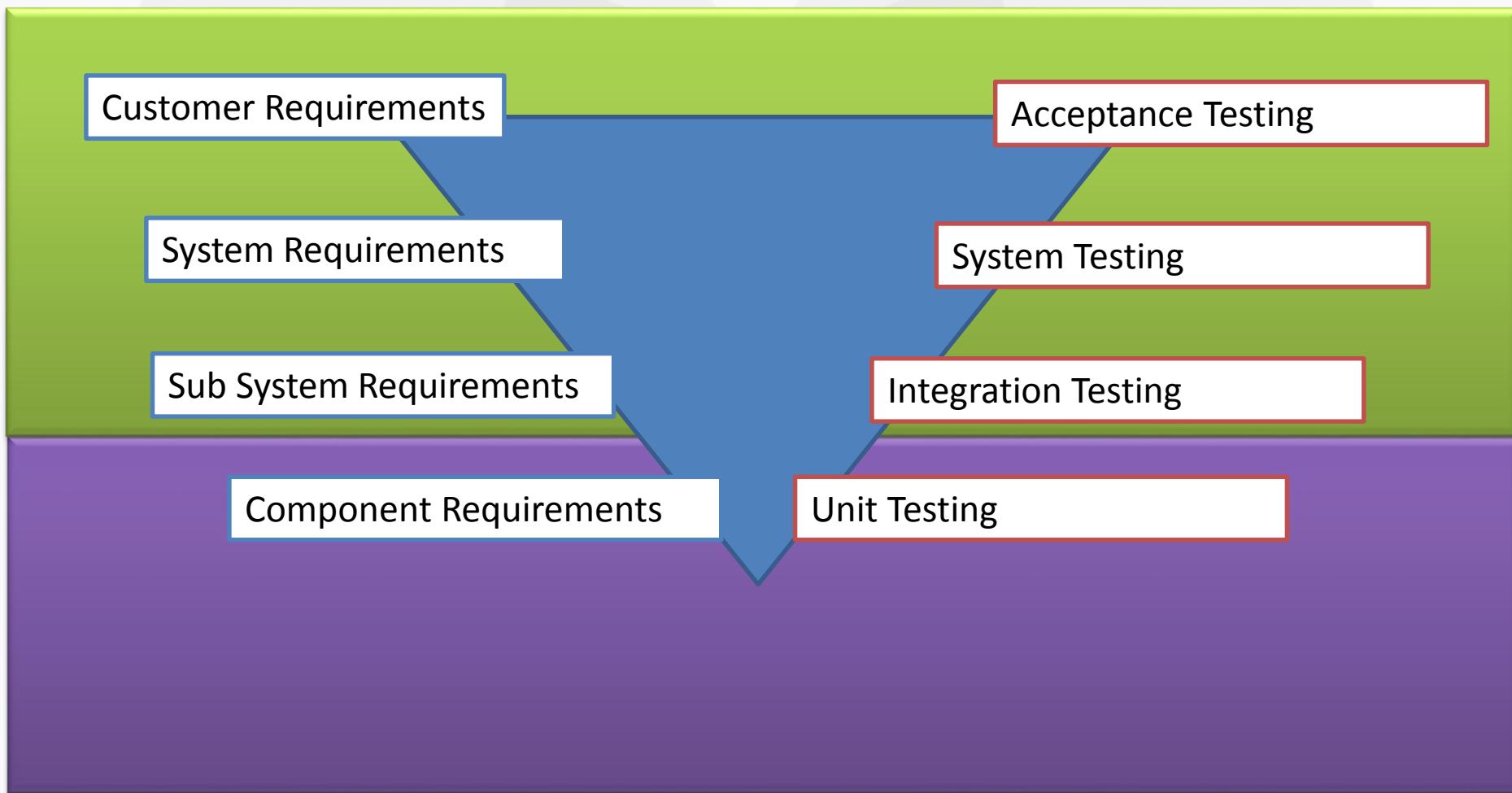
Discussion

- Do some study about Ubuntu release testing process?

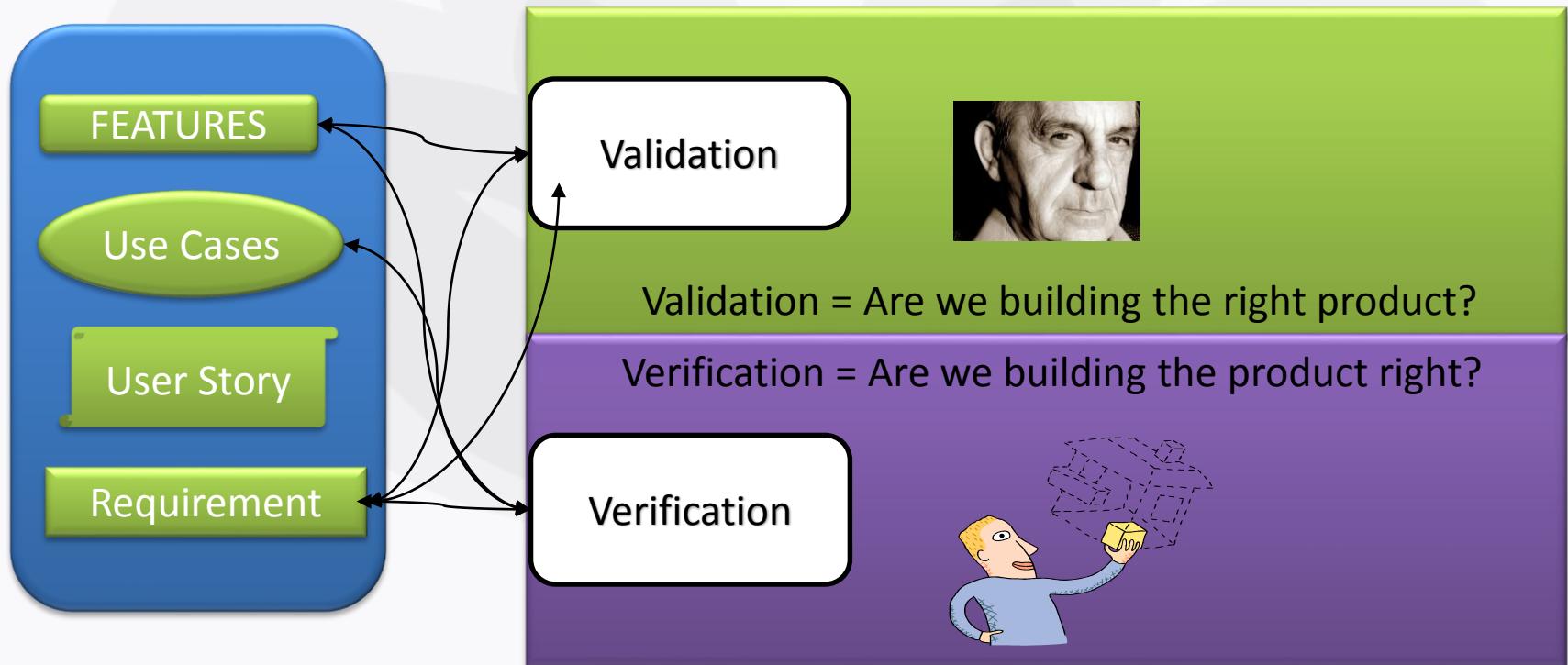
V-Model for testing



Developer vs Tester??

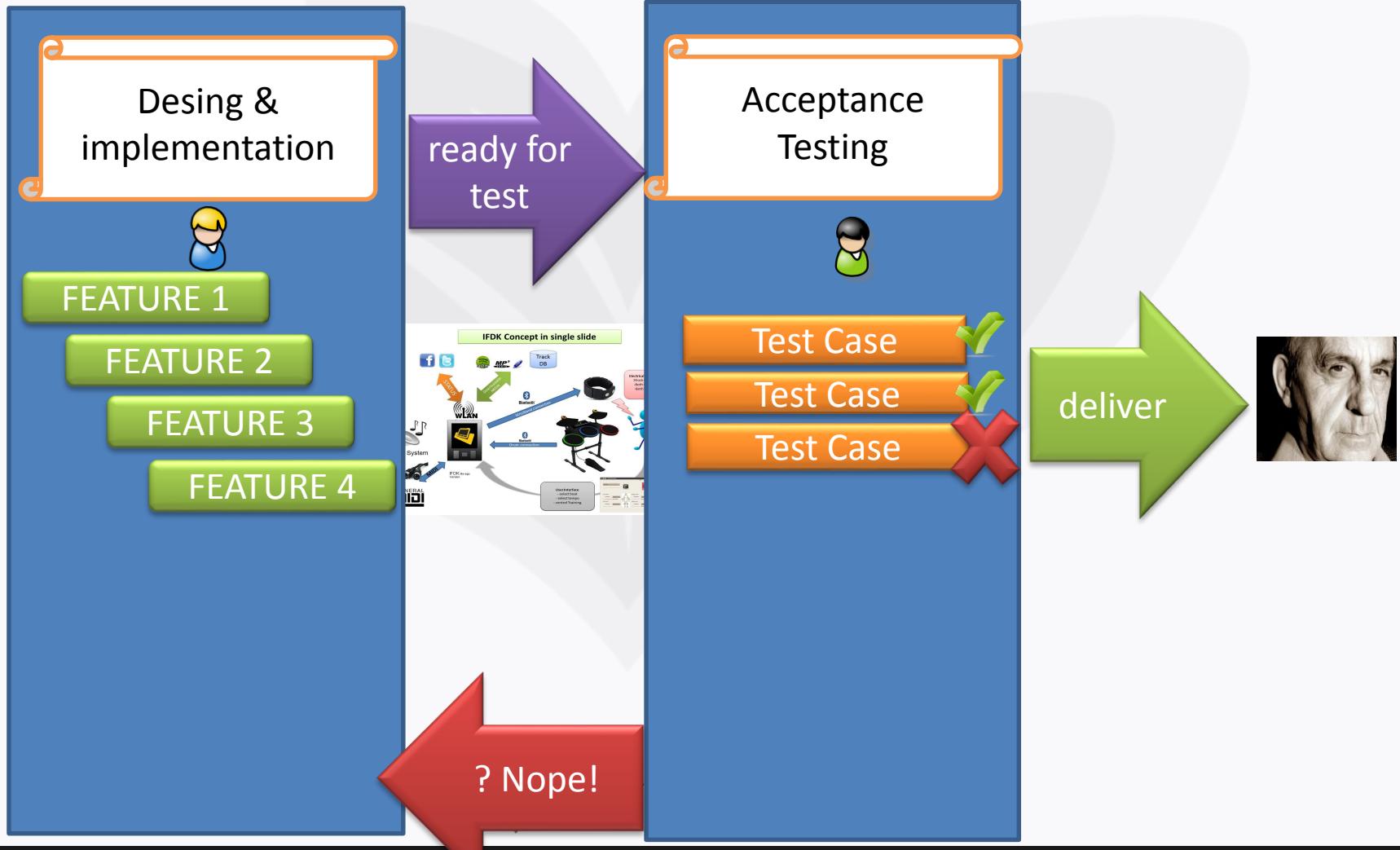


Verification & Validation



Validation VS verification?

Testing in brief?



Testing Orientation

Black Box Testing

Grey Box Testing

White Box Testing

"System Testing"- perspective

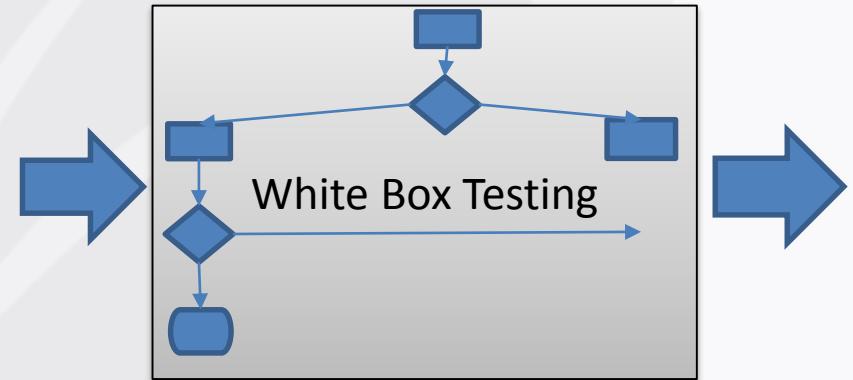


"Code level testing"- perspective

http://en.wikipedia.org/wiki/Software_testing

Testing Orientation

"System Testing"- perspective



"Code level testing"- perspective

http://en.wikipedia.org/wiki/Software_testing

Development Process (Waterfall)

Requirement Gathering

Design

Implementation

Verification

Validation

Maintenance

Milestone 1

Milestone 2

Milestone 3

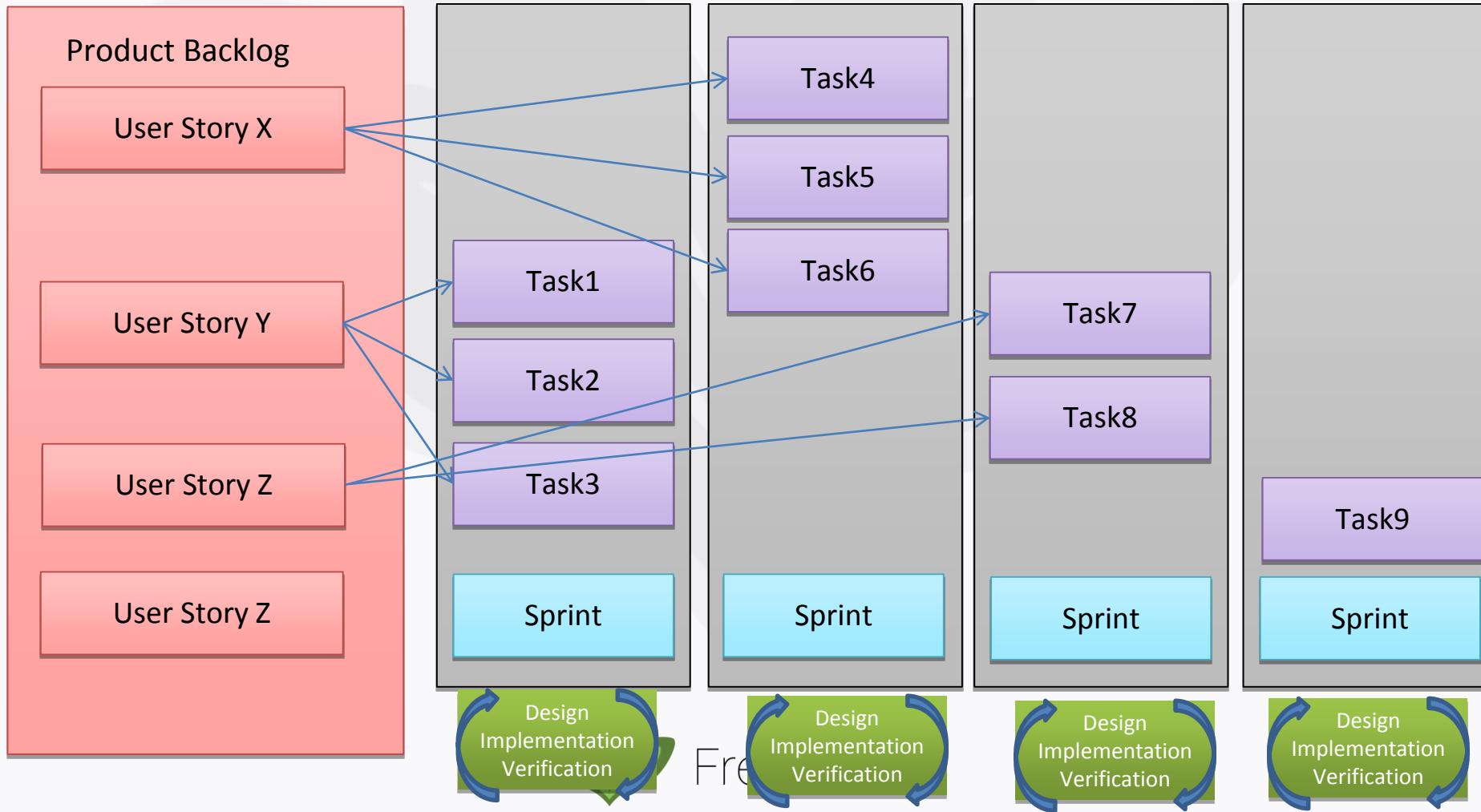
Milestone 4

Milestone 5



Free**Nest**

Development Process (Agile)



Testing levels

- We take branding seriously, The FreeNest Brand is our pride and joy.
- It is the message we want to send to the developer teams, customers and stakeholders.
- These guidelines must be followed in all FreeNest's print and electronic communications, marketing and informing.
- Documentation defines logo guidelines, typography and the use of colors.

How to test?

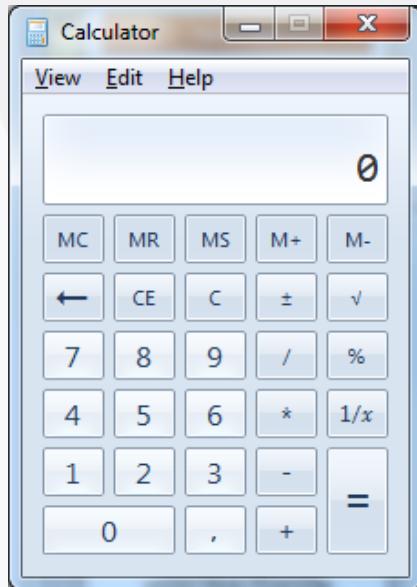
```
void
mp_tan(const MPNumber *x, MPAngleUnit unit, MPNumber *z)
{
    MPNumber cos_x, sin_x;

    /* Check for undefined values */
    mp_cos(x, unit, &cos_x);
    if (mp_is_zero(&cos_x)) {
        /* Translators: Error displayed when tangent value is undefined */
        mperr_("Tangent is undefined for angles that are multiples of π (180°) from π/2 (90°)");
        mp_set_from_integer(0, z);
        return;
    }

    /* tan(x) = sin(x) / cos(x) */
    mp_sin(x, unit, &sin_x);
    mp_divide(&sin_x, &cos_x, z);
}
```

What should be tested? How?

How to test?



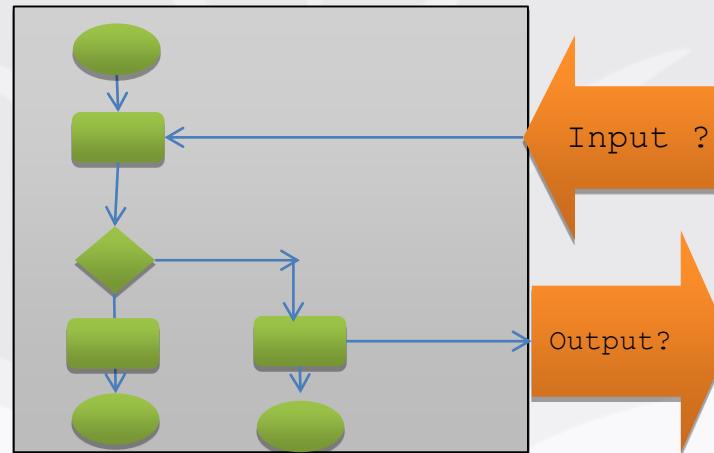
What should be tested?

How ?

Verification of implementation

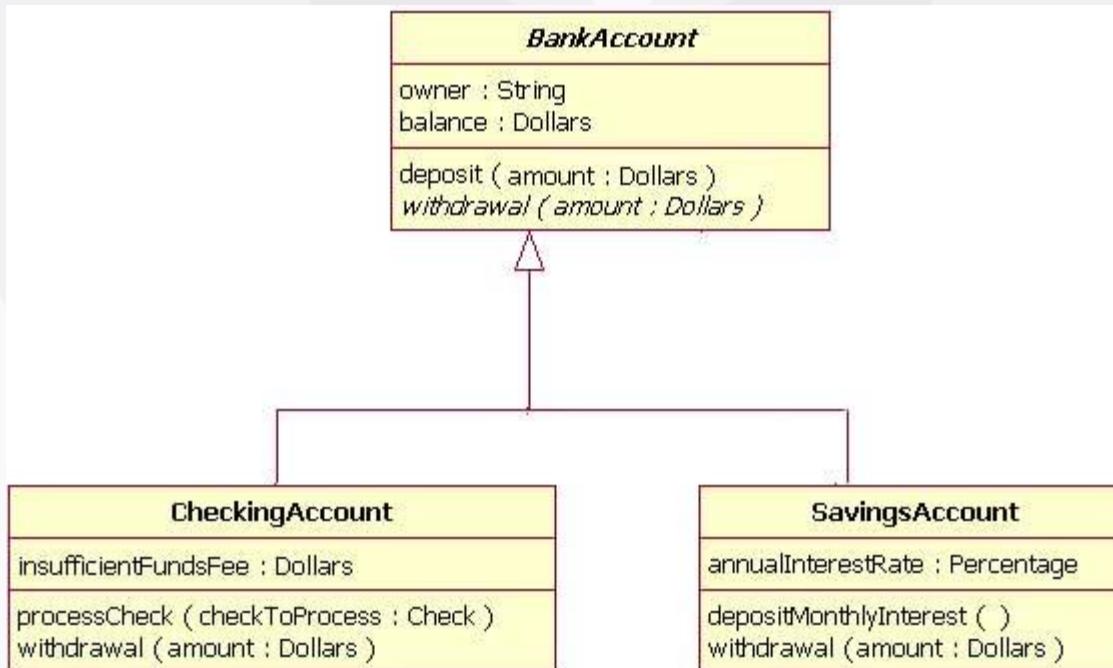
- Unit Testing
- Static Analyze
- Dynamic Analyze

What should be tested?



Free**Nest**

Class model ?



[http://www.ibm.com/developerworks/rational/library
/content/RationalEdge/sep04/bell/](http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/)

xUnit Framework

xUnit Framework

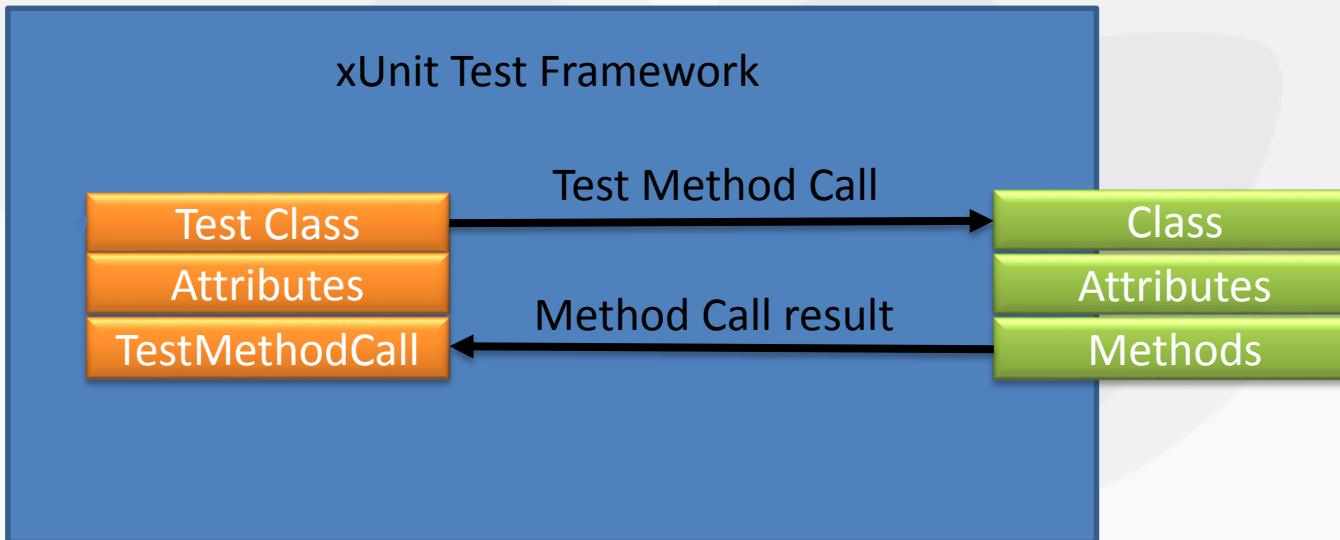
Implementation



Component/ Unit Testing



Developer



a=1; b= 2

C=Class.TestCountValues(1,2)

C <> 3 = Fail

C = 3 PASS

MethodCountValues(int x ,int y)
z = x+y+1
Return z



Free**Nest**

Test Driven Development?



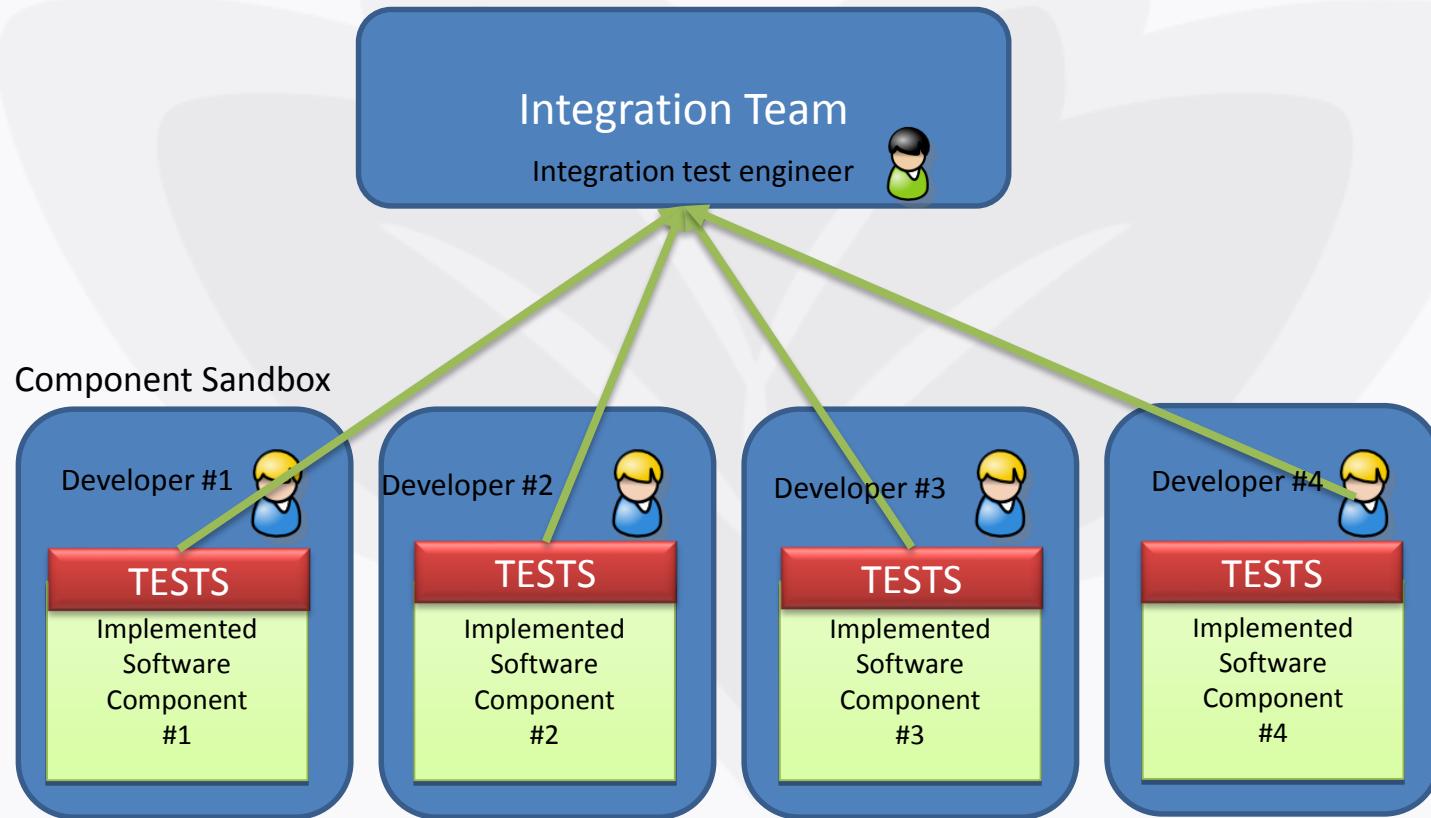
DEFINE
TEST
CASE
FIRST!!!

IMPLEMENT
CODE
AGAINST
TESTS



Free**Nest**

Ideal project team and unit testing



Integration Testing



Free**Nest**

How to test ?

The screenshot shows a web application for calculating tax. At the top, there's a header bar with a back button, a search bar containing 'verohallinto', and a URL 'http://prosentti.vero.fi/veropros_tietojen_syotto2011.asp'. Below the header, the page title is 'Etusivu > Vero%laskuri'.

Vero%laskuri 2011

Henkilötiedot

Kotikunta 31.12.2010	<input type="button" value="Valitse kotikunta"/>
Seurakuntaan kuuluminen 31.12.2010	<input type="button" value="Valitse"/>
Ikä	<input type="button" value="18 - 53 vuotta"/>
Alaikäisten lasten lukumäärä alijäämähyytyksen lapsikorotusta varten	<input type="button"/>
Lasten lukumäärä elatusvelvollisuusvähennystä varten	<input type="button"/>
Invalidiprosentti	<input type="button"/>
Vuoden 1982 invalidiprosentti	<input type="button"/>
Muulla kuin Suomessa vakuutettu	<input type="checkbox"/>

Ennakonpidätyskseen alaiset ansiotulot

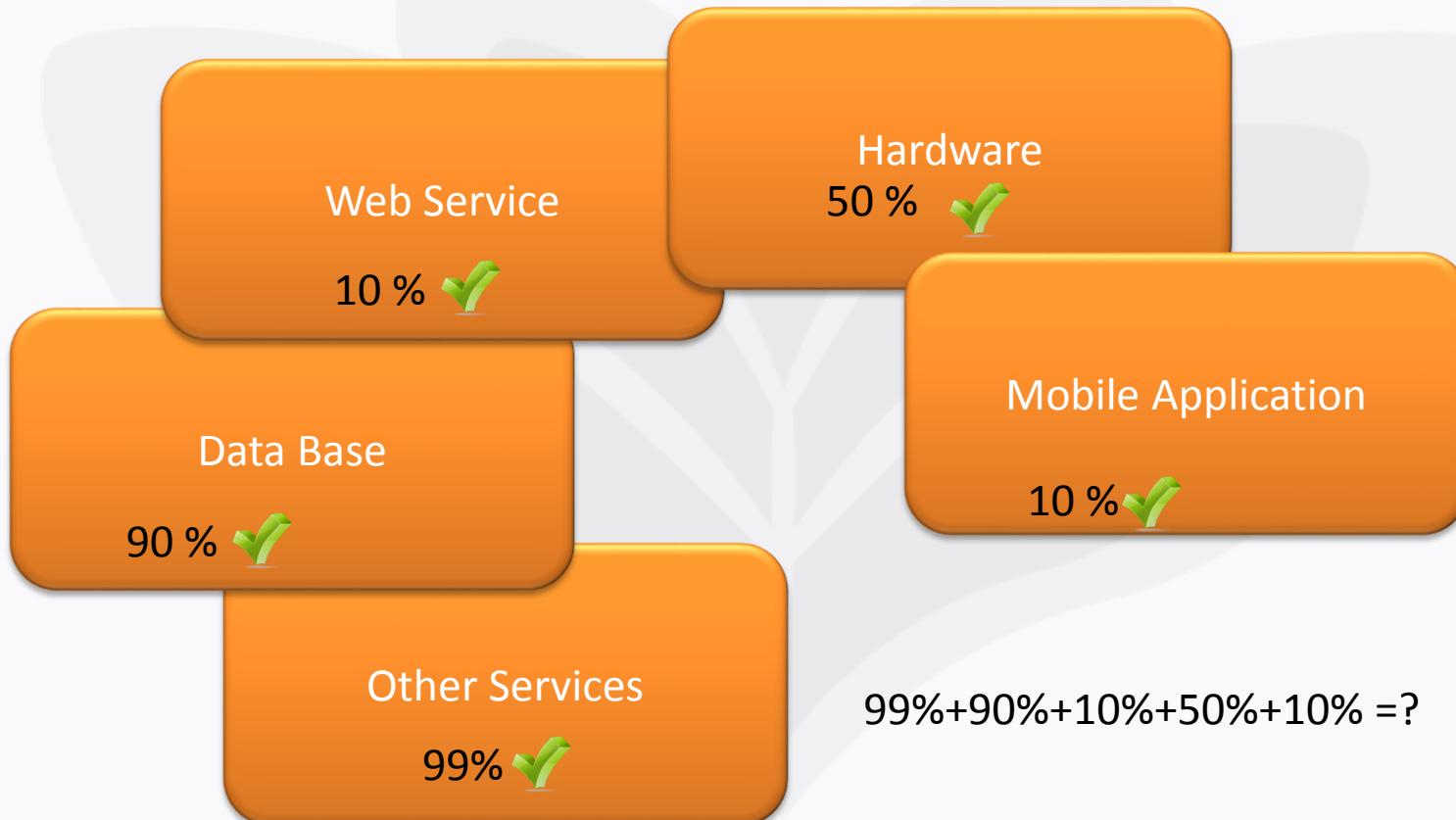
	Arvio koko vuoden bruttotuloista	Saatu bruttoton vuoden alusta	Ennakonpidäys vuoden alusta
Päätoiminen palkkatulo luontoisetuineen	<input type="button"/>	<input type="button"/>	<input type="button"/>
Sivutulot luontoisetuineen	<input type="button"/>	<input type="button"/>	<input type="button"/>
Sosiaali- ja muut etuudet	<input type="button"/>	<input type="button"/>	<input type="button"/>
Merityö tulot luontoisetuineen	<input type="button"/>	<input type="button"/>	<input type="button"/>
Perhehoitajan palkkio, omaishoidon tuki	<input type="button"/>	<input type="button"/>	<input type="button"/>
Ennakonpidätyskseen alainen työkorvaus	<input type="button"/>	<input type="button"/>	<input type="button"/>
Käyttökorvaus	<input type="button"/>	<input type="button"/>	<input type="button"/>
Kansaneläke ja takuueläke	<input type="button"/>	<input type="button"/>	<input type="button"/>

What should be tested?

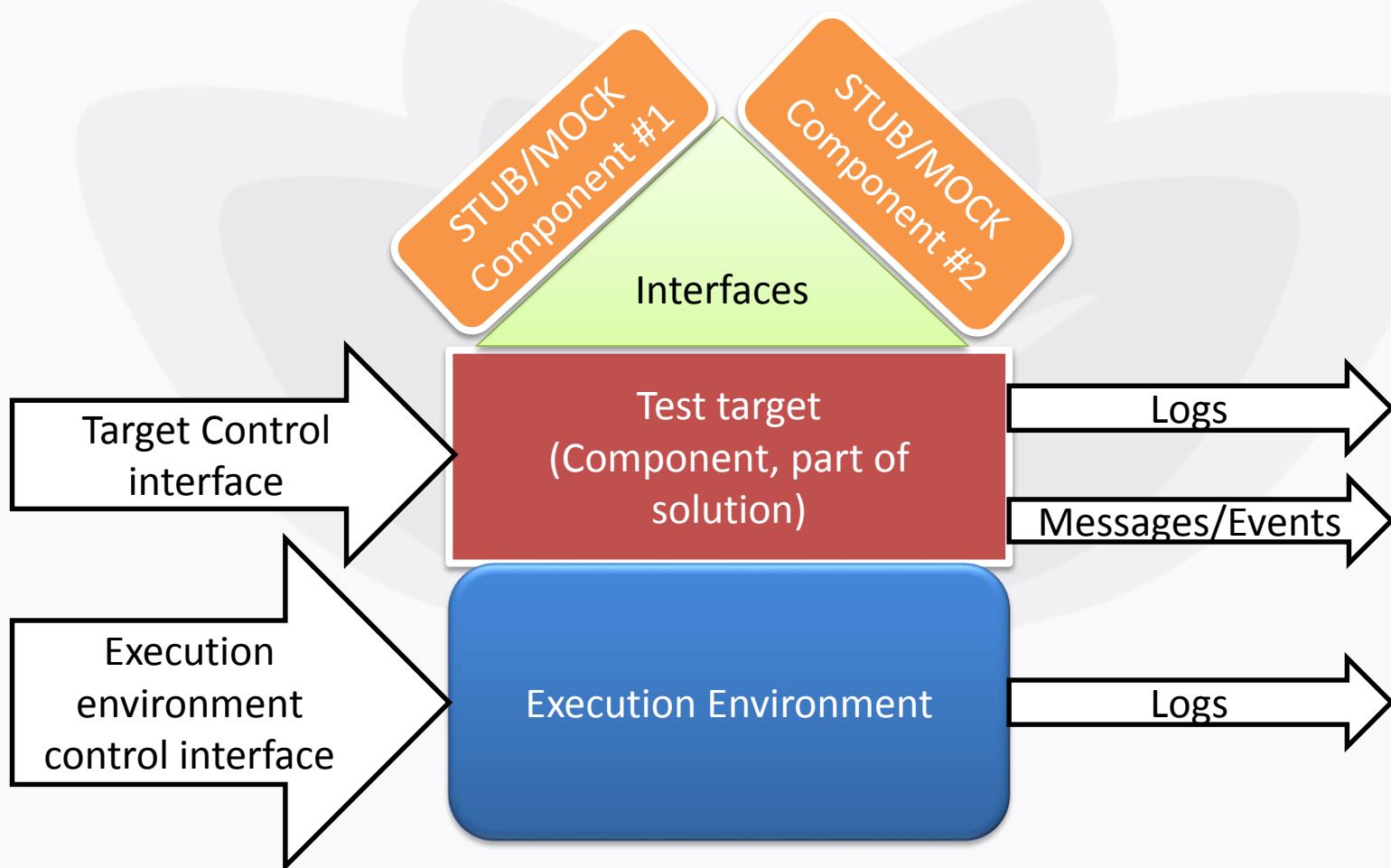
How ?

http://prosentti.vero.fi/veropros_tietojen_syotto2011.asp

Integrate early! Avoid Big Bang

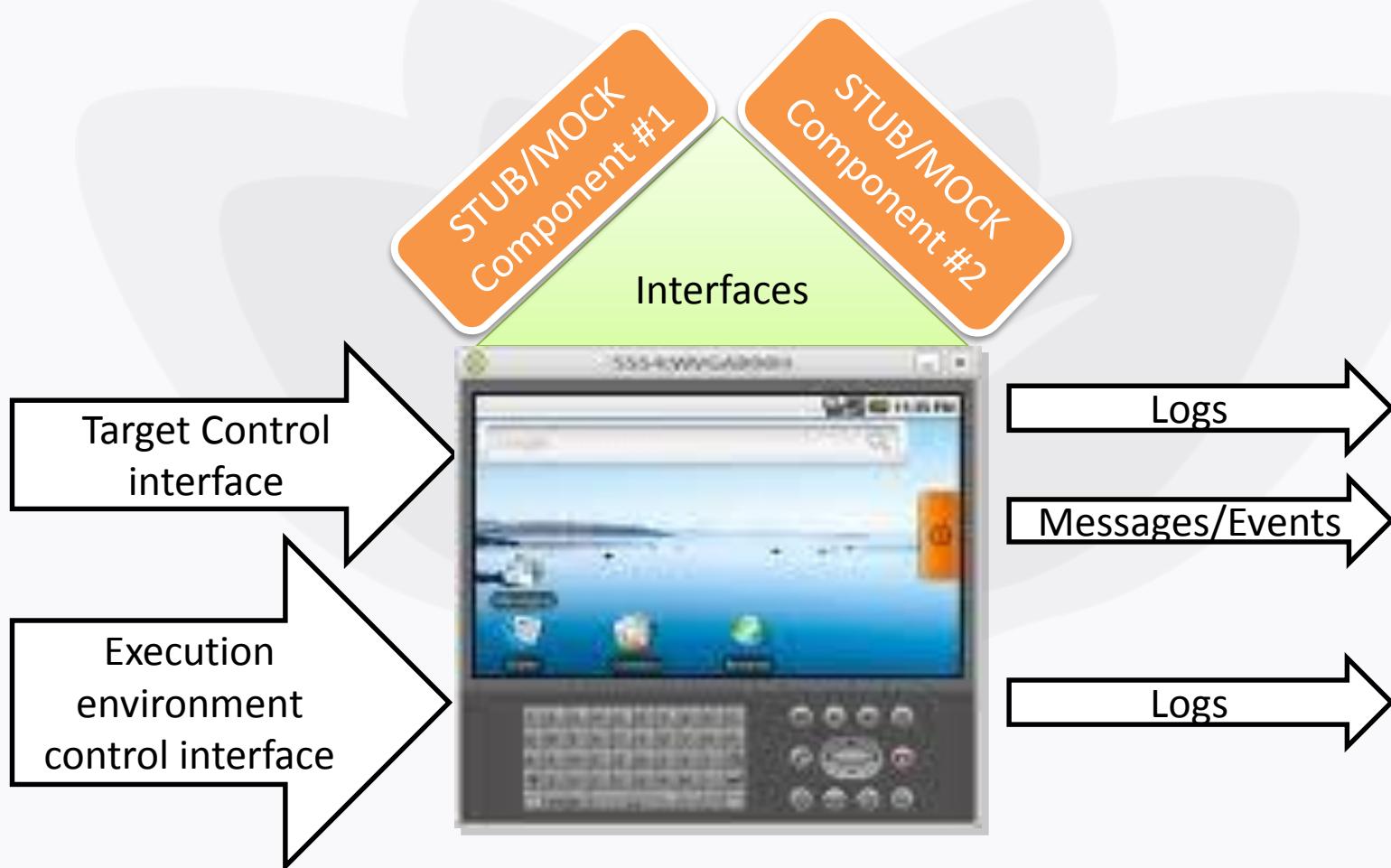


Integration Testing with stubs



FreeNest

Integration Testing with stubs



FreeNest

How to test?



What should be tested?

How ?

System Testing in Large

http://prosentti.vero.fi/veropros_tietojen_syotto2011.asp

Etusivu > Vero%laskuri

Vero%laskuri 2011

Henkilötiedot

Kotikunta 31.12.2010	<input type="button" value="Valitse kotikunta"/>
Seurakuntaan kuuluminen 31.12.2010	<input type="button" value="Valitse"/>
Ikä	<input type="button" value="18 - 53 vuotta"/>
Alaikäisten lasten lukumäärä alijäämähyytyksen lapsikorotusta varten	<input type="button"/>
Lasten lukumäärä elatusvelvollisuusvähennystä varten	<input type="button"/>
Invalidiprosentti	<input type="button"/>
Vuoden 1982 invalidiprosentti	<input type="button"/>
Muulla kuin Suomessa vakuutettu	<input type="button"/> <input type="checkbox"/>

.....

Ennakonpidätysken alaiset ansiotulot	Arvio koko vuoden bruttotuloista	Saatu bruttotulo vuoden alusta	Ennakonpidäys vuoden alusta
Päätoimen palkkatulo luontoisetuinne	<input type="button"/>	<input type="button"/>	<input type="button"/>
Sivutulot luontoisetuinne	<input type="button"/>	<input type="button"/>	<input type="button"/>
Sosiaali- ja muut etuudet	<input type="button"/>	<input type="button"/>	<input type="button"/>
Merityötulot luontoisetuinne	<input type="button"/>	<input type="button"/>	<input type="button"/>
Perhehoitajan palkkio, omaishoidon tuki	<input type="button"/>	<input type="button"/>	<input type="button"/>
Ennakonpidäyskenenalainen työkorvaus	<input type="button"/>	<input type="button"/>	<input type="button"/>
Käyttökorvaus	<input type="button"/>	<input type="button"/>	<input type="button"/>
Kansanelake ja takuuelake	<input type="button"/>	<input type="button"/>	<input type="button"/>

What should be tested?

How ?

System Testing in Large

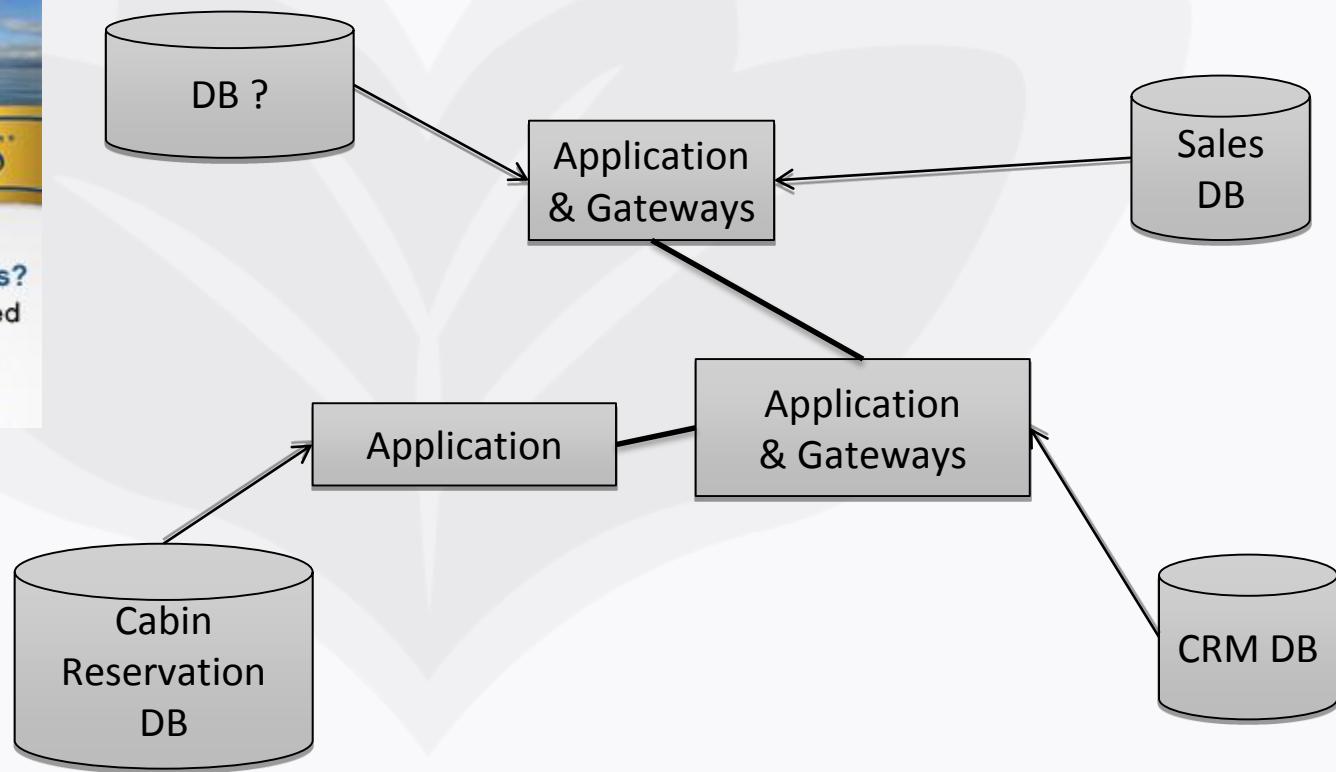


Excited about Royal Princess?
Then you'll love this detail-packed
PDF overview!

[View Now](#)

What should be tested?

How ?



FreeNest

Acceptance Testing



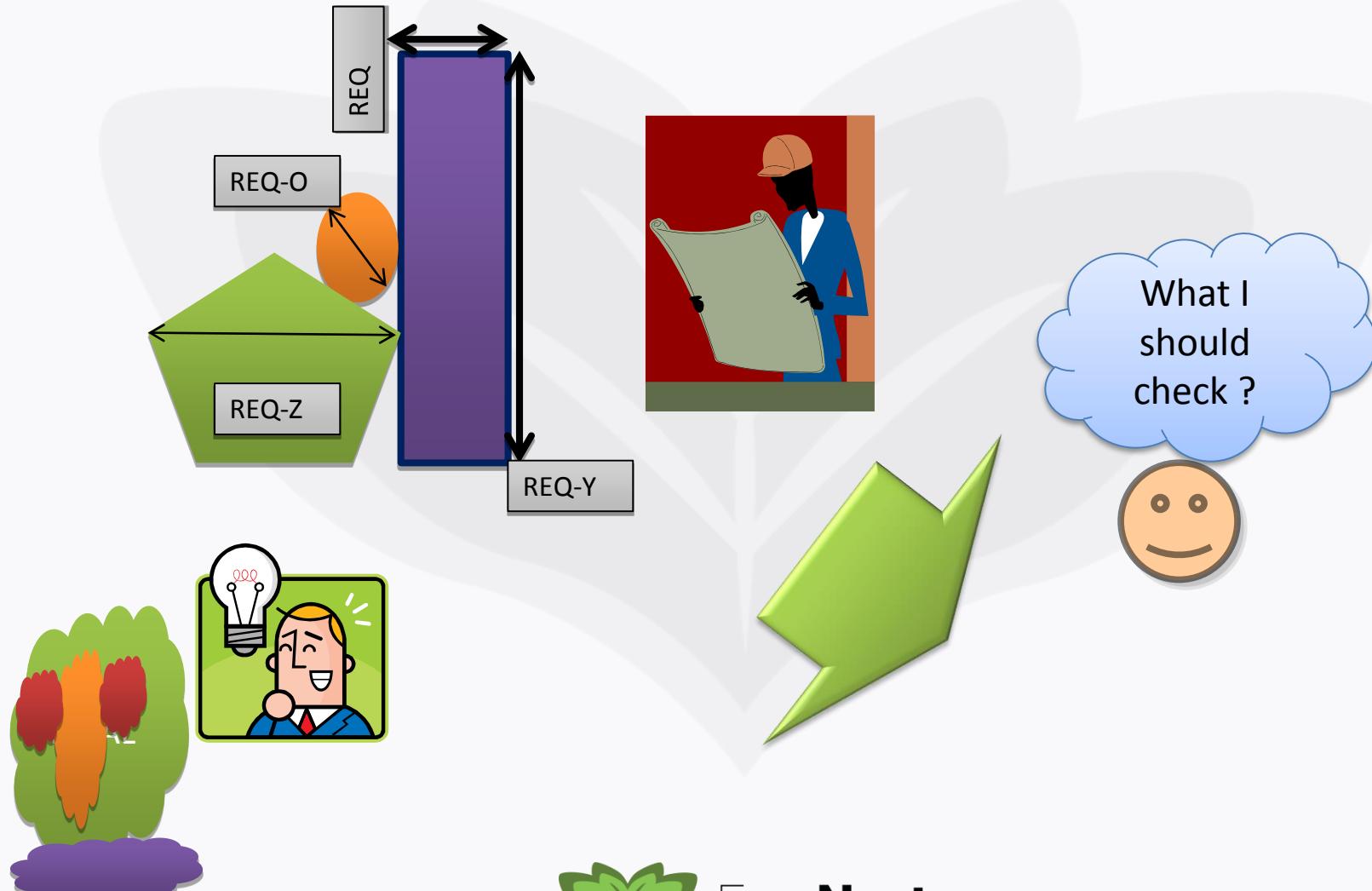
What should be tested before so customer could be so happy ?

How ?

Test design & execution

- Discuss about reasons for test design?
- Why we need to do design?
- Stupid work 😊 ! I wan't to progress!??

What is test design?



Test Case ?

Add Information about case

Verify what?
Using configuration?
With tools?

Define pre-state
Define Steps
Define end-state

What is verdict?

- . Test Case Name:
- . Test Case Id:
- . Test Case owner/writer:
- . Date
- . comments

- . Verify drum track player pause mode functionality.
- . Do this with IFDK software release X and playing song "Show must go on by Freddy Mercury"
- . Test should be done using android emulator environment and using your hands, ears and eyes"

- . Pre State:
 - . Android emulator is running
 - . Release X is installed on emulator
- . Test Case Steps:
 1. Open drum kit player application
 2. Select song "Show must go on"
 3. Start to play
 4. Press Pause and check song is paused
 5. Check memory usage from system application
 6. Press Play
 7. jump to 4 several time (<10)
 8. Listen song to the end
 9. Exit player using "exit button"

- . End State:
 - . IFDK Kit in main screen mode

- . If Pause is working result is PASS. If Pause mode failed result is FAIL



Free

Why we need test design ???

- Stupid work! This takes ages! This Test Case documentation is old as soon I have changed some implementation? Why you need to do so hard documentation? Give me a one good reason!

Agile Thinking?

- We have to automate all tests!! No sense to create test case documentation ?
- Who does automation without a design?

Checklist vs Test Case?

Check UI is working

Working?

Check color change

Working?

Check Counter value after 50 logins

Working?

Check disable mode for counter

Working?

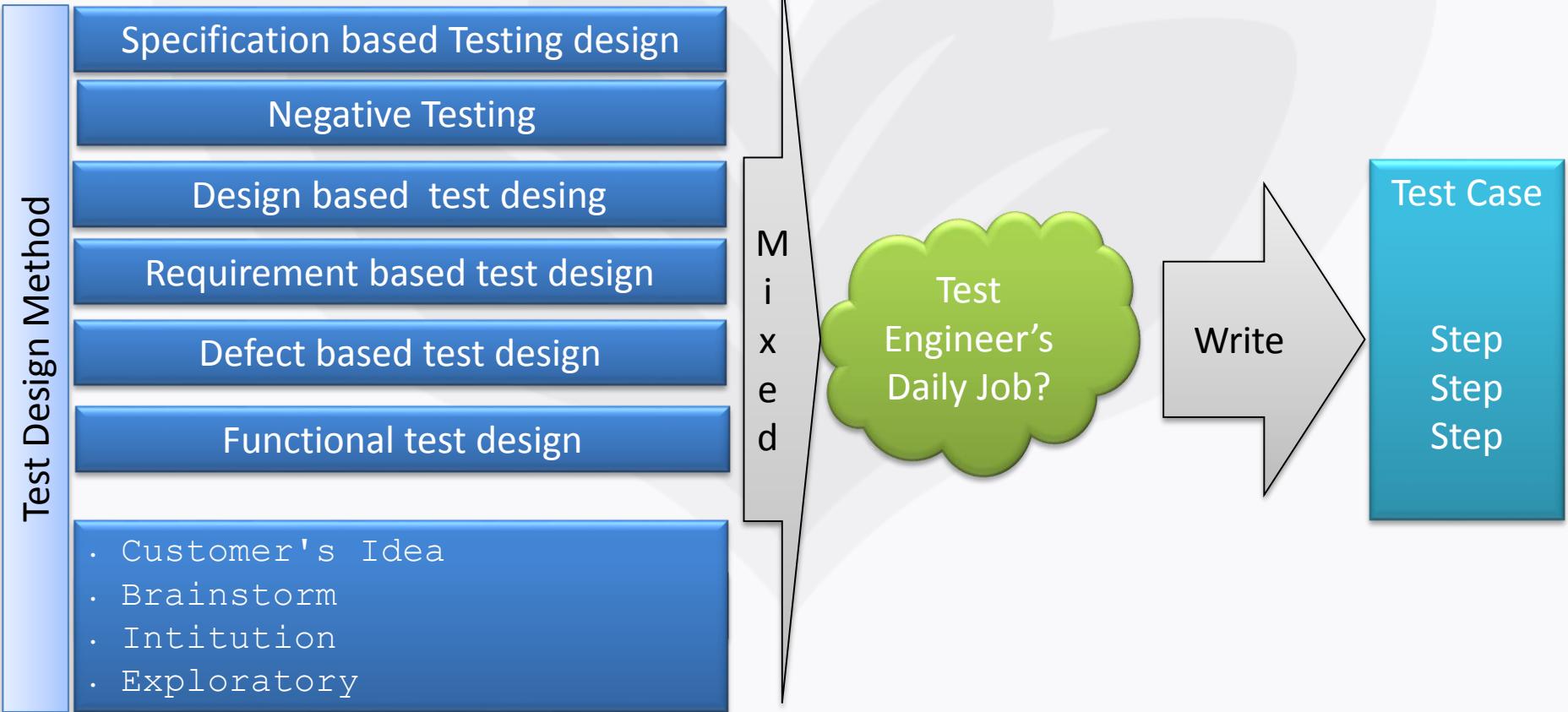
Checklist can be working great in small team! ☺

What happens if team is disbanded to other projects? And you are new maintainer for this project?

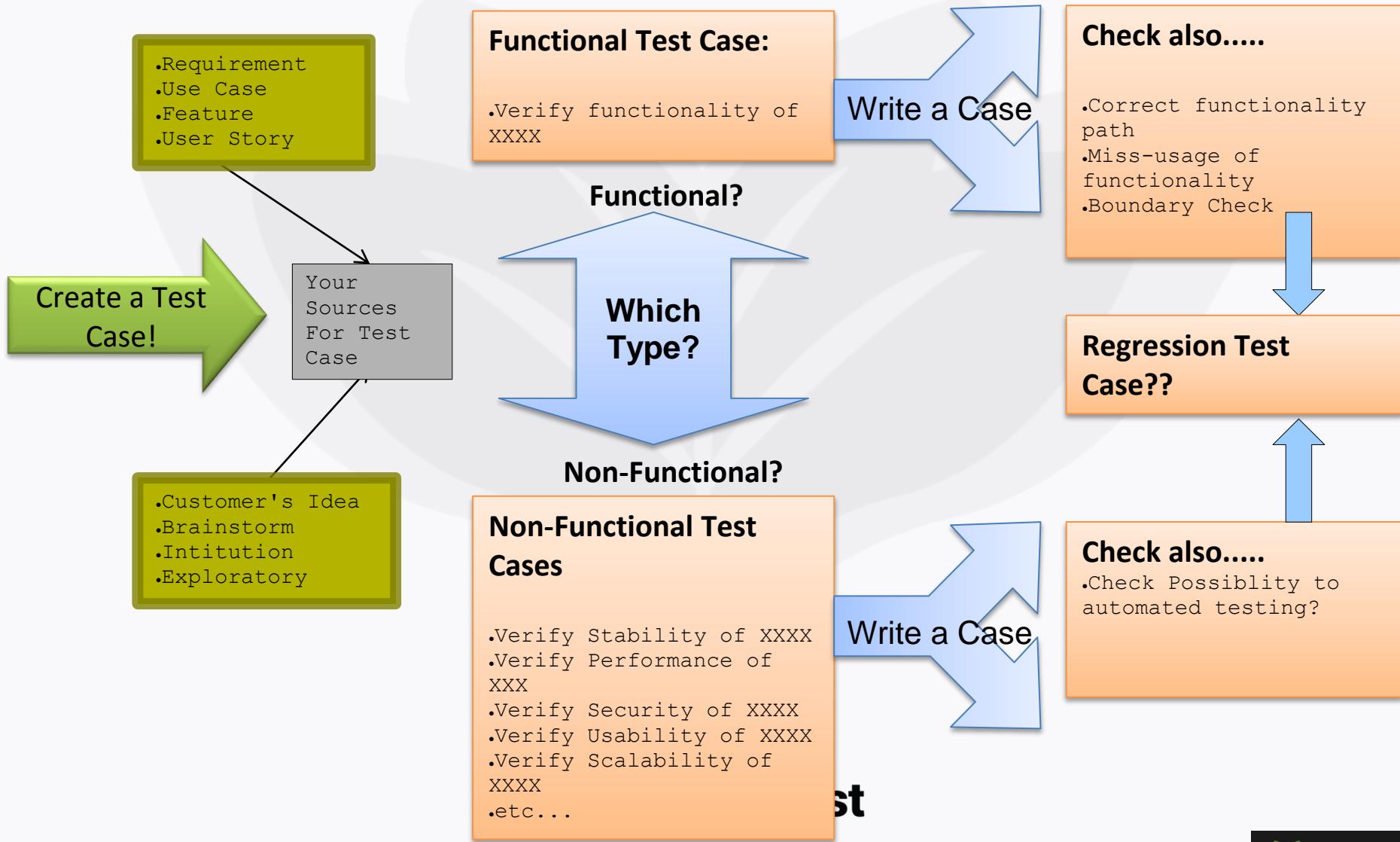


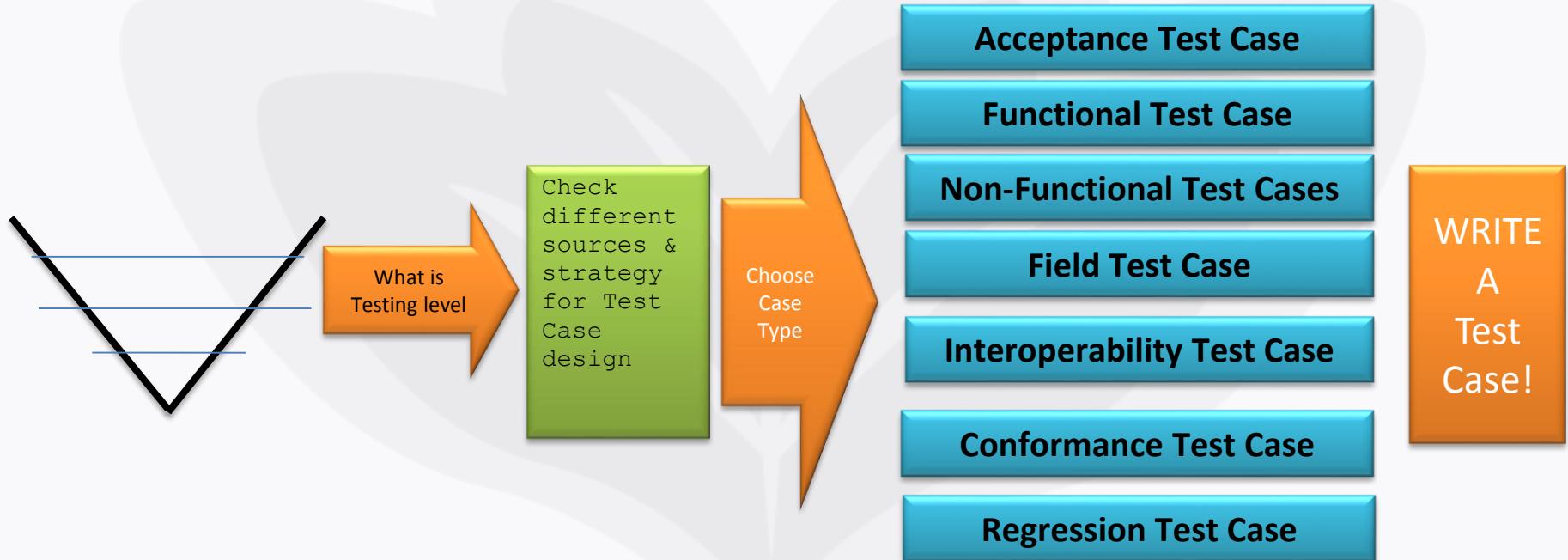
Free**Nest**

Sources for test design

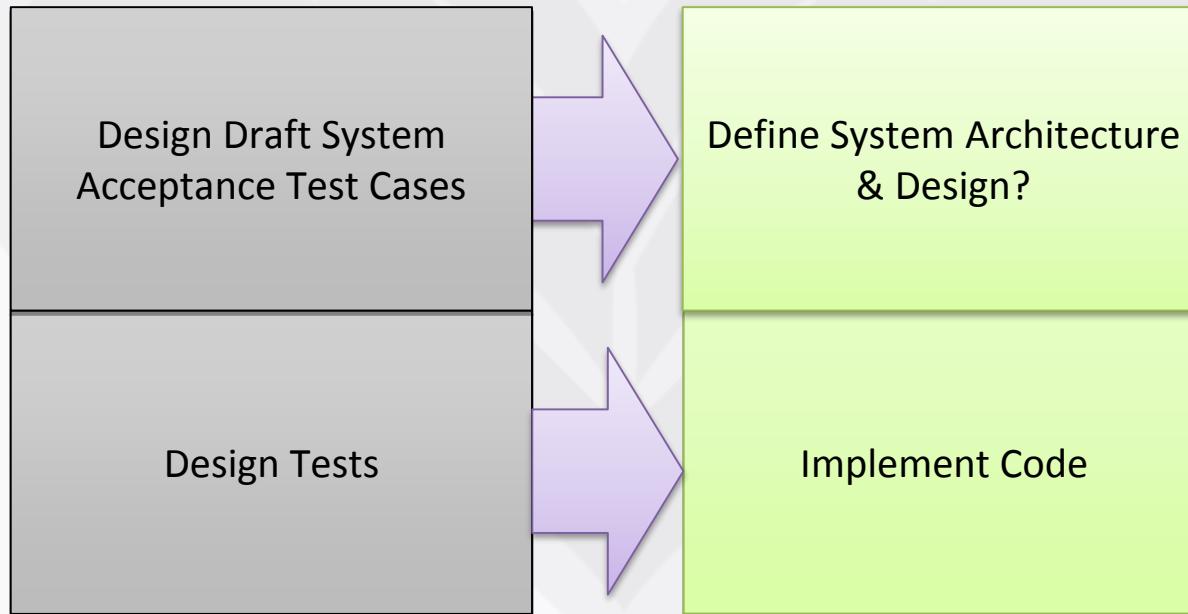


Mechanical route to design





Test Driven Development in all testing levels?

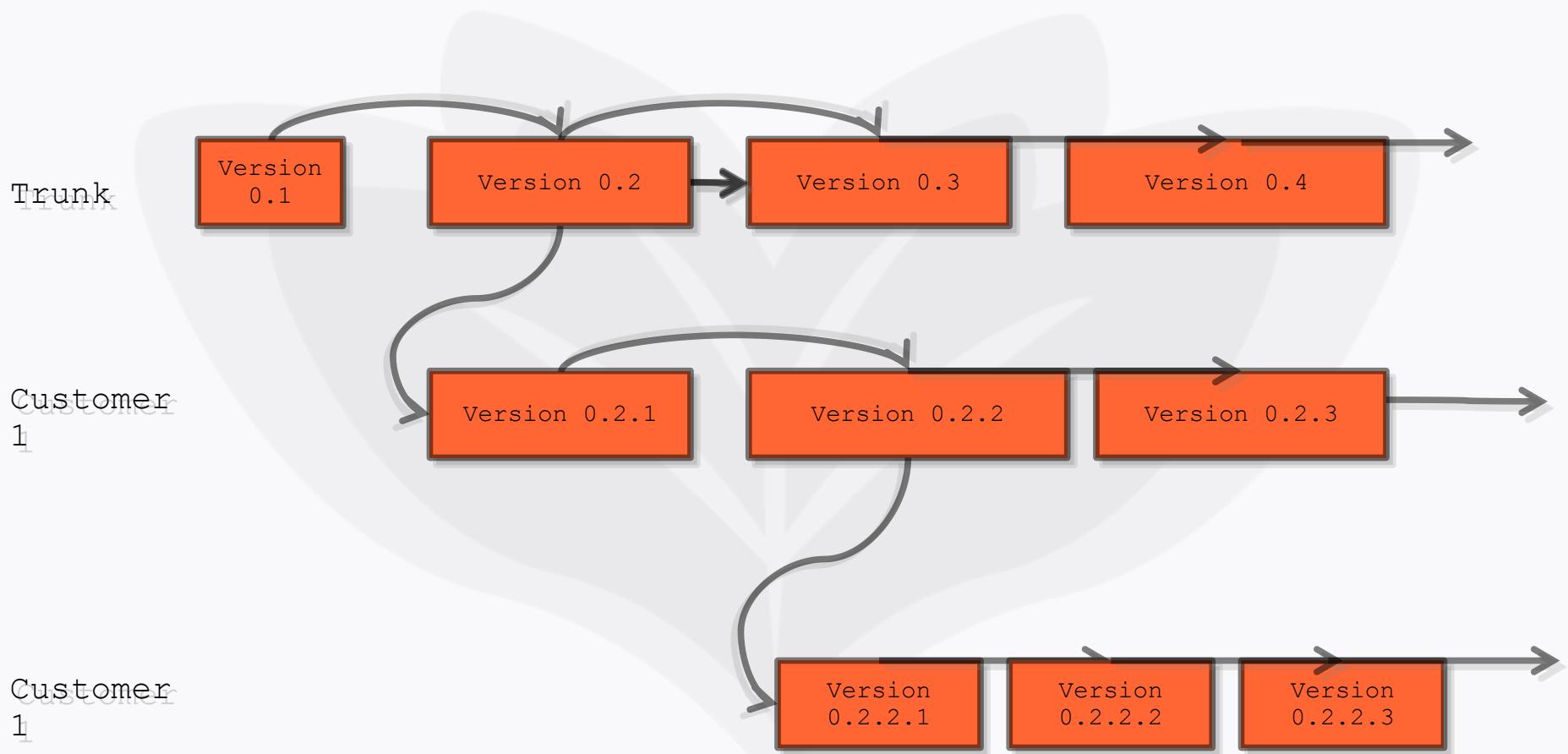


Free**Nest**

Release Management

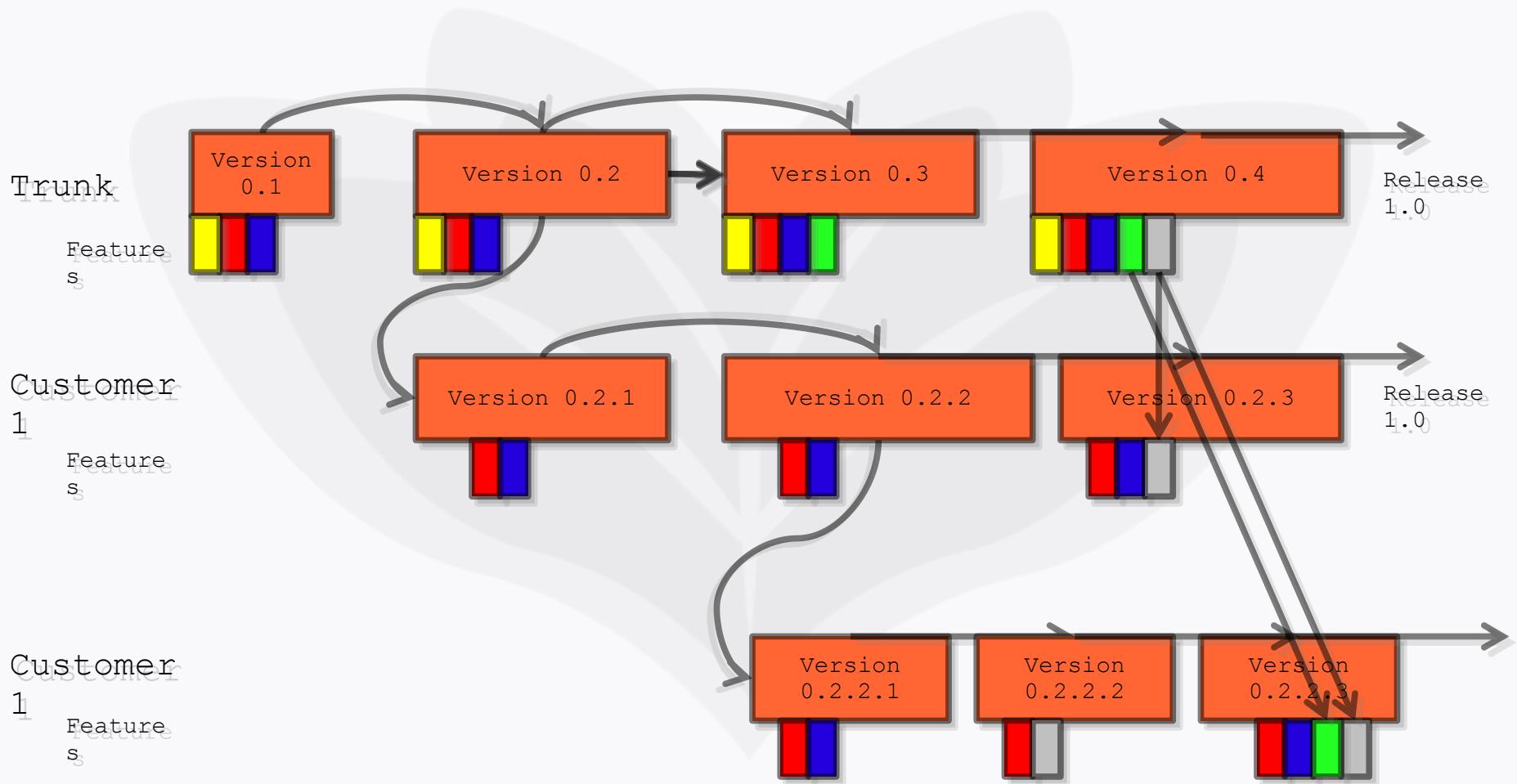
- http://en.wikipedia.org/wiki/Configuration_management
- http://en.wikipedia.org/wiki/Release_management

Release Management



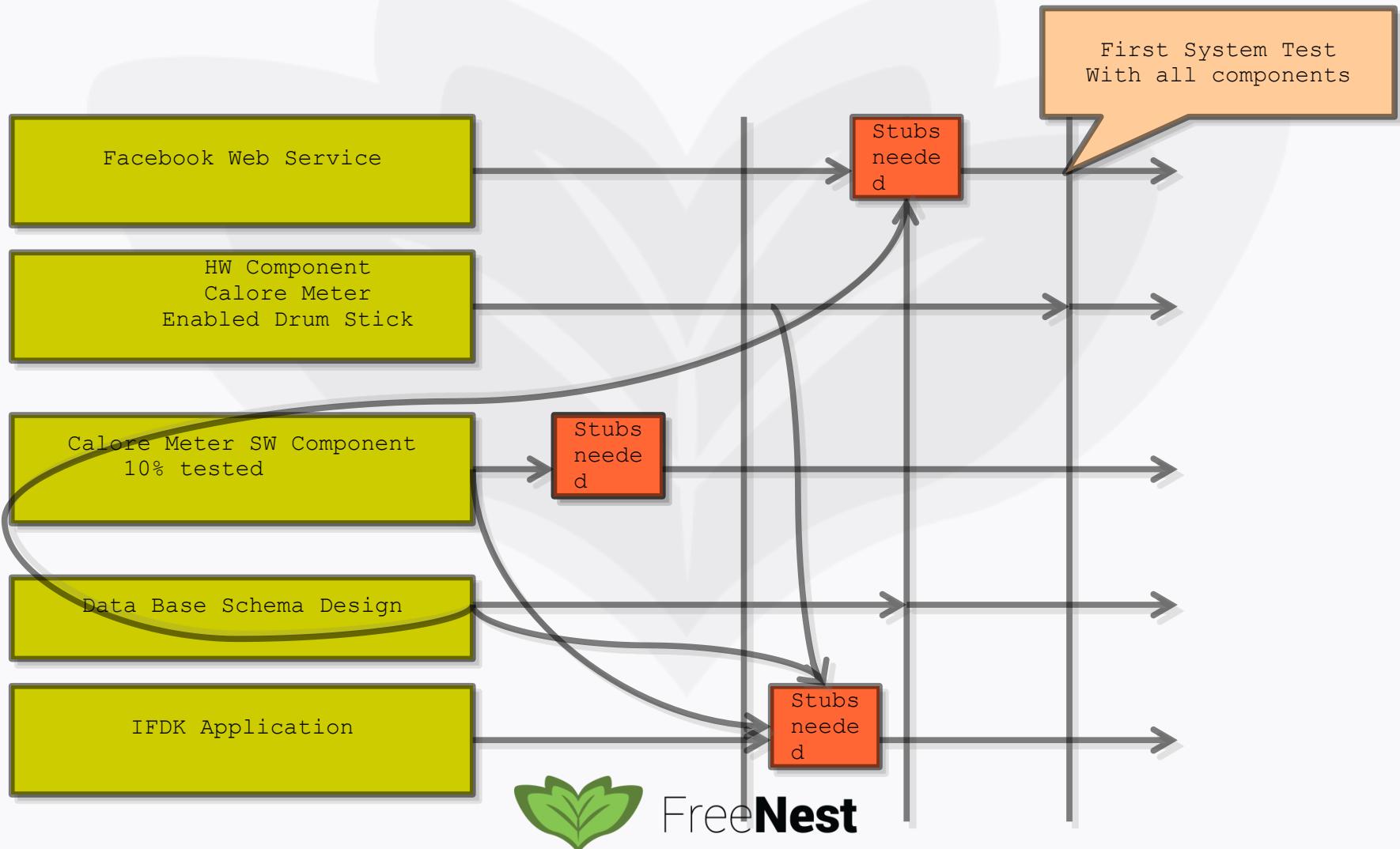
Free**Nest**

Release & Configuration Management



Free**Nest**

Release Management and integration testing



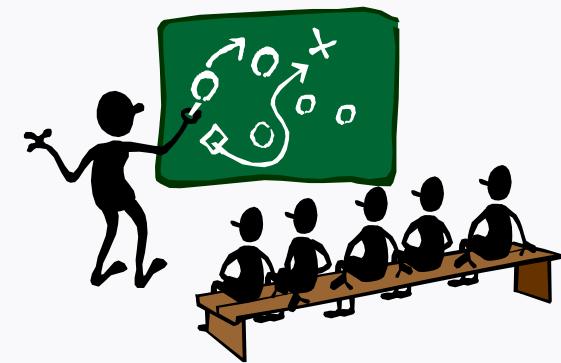
Test Planning & Management

Why Test Management is needed ?

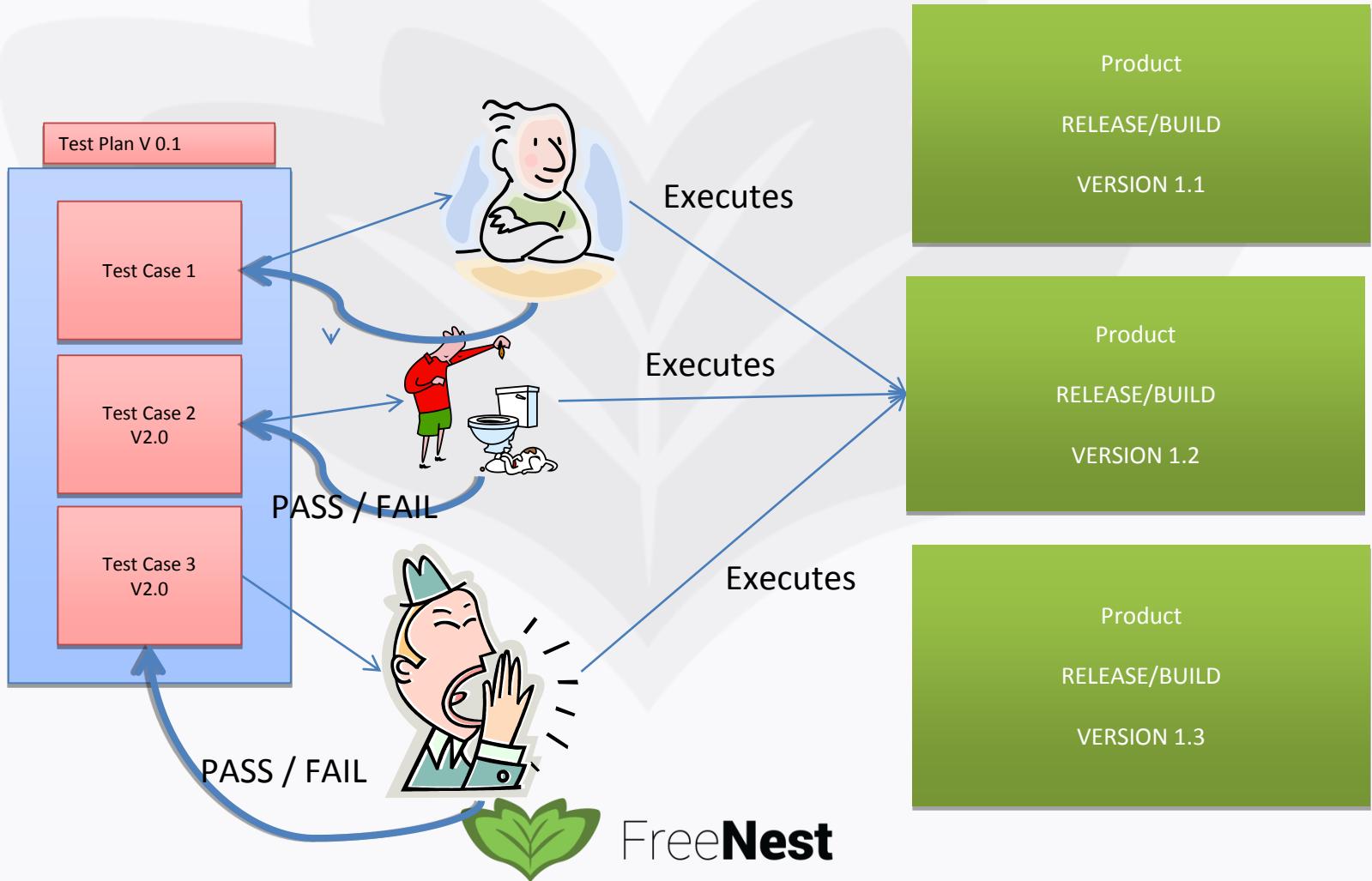


Free**Nest**

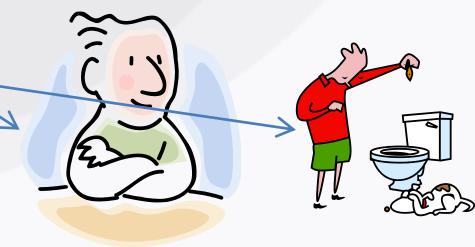
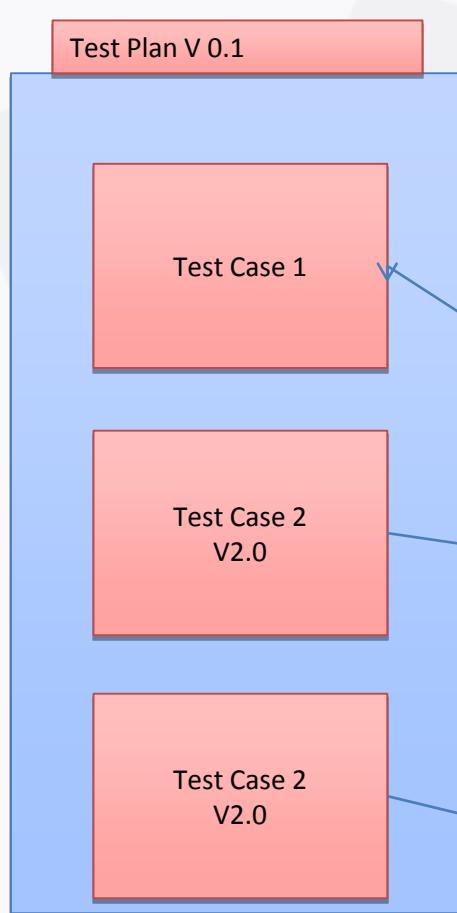
What should be tested first?



What is a test target

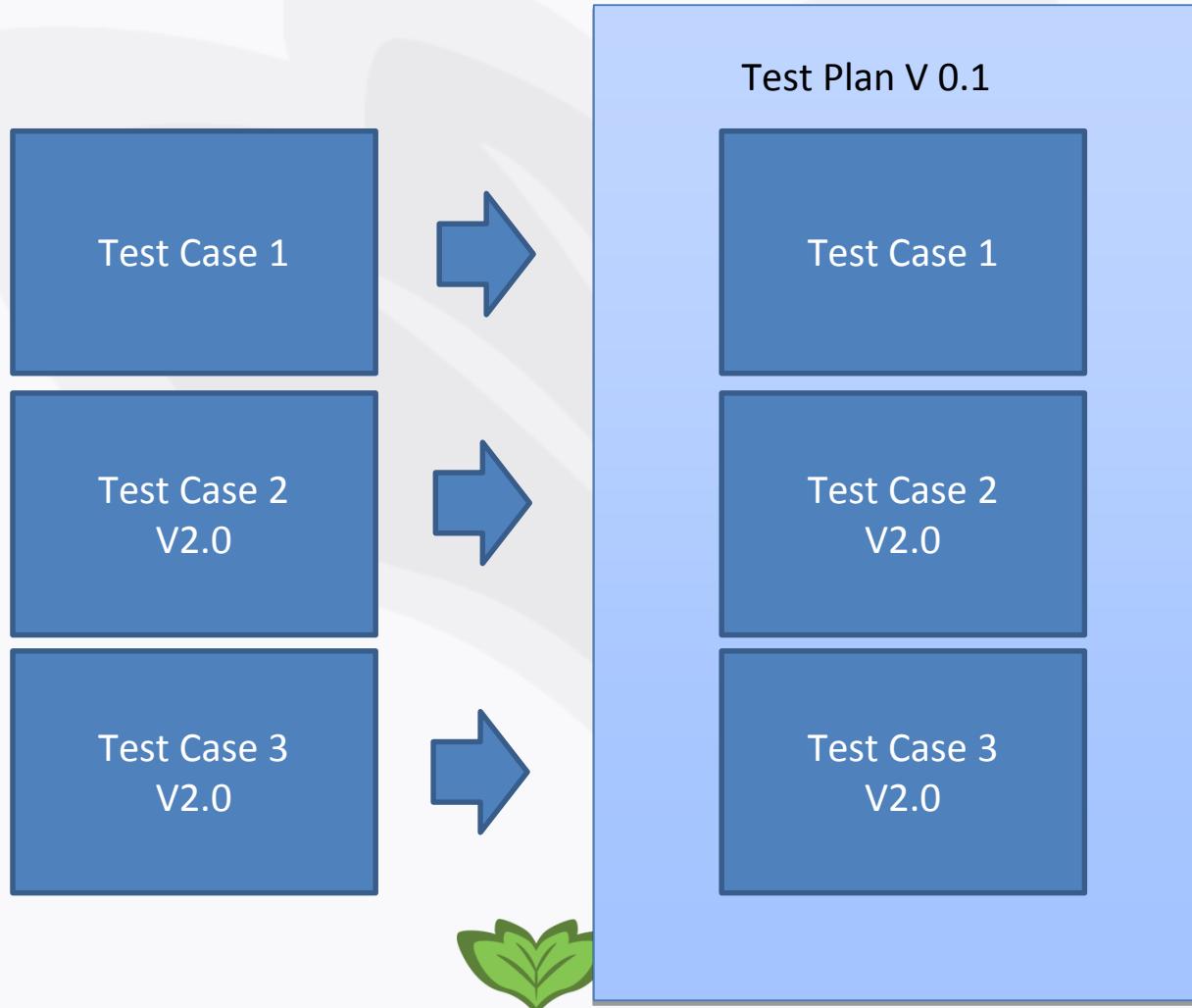


Who executes test cases and when

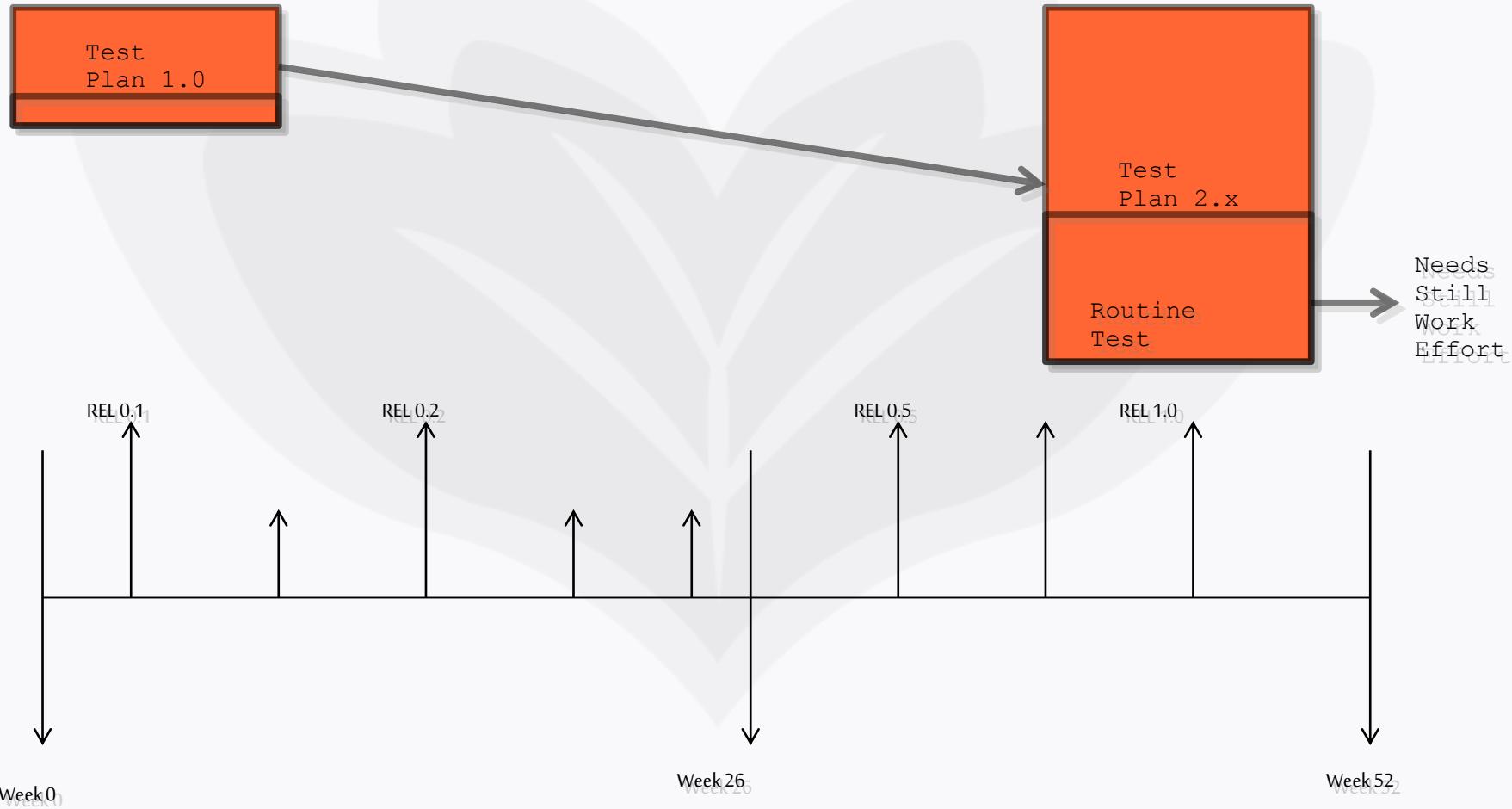


Free**Nest**

Group of Test Cases = Test Plan

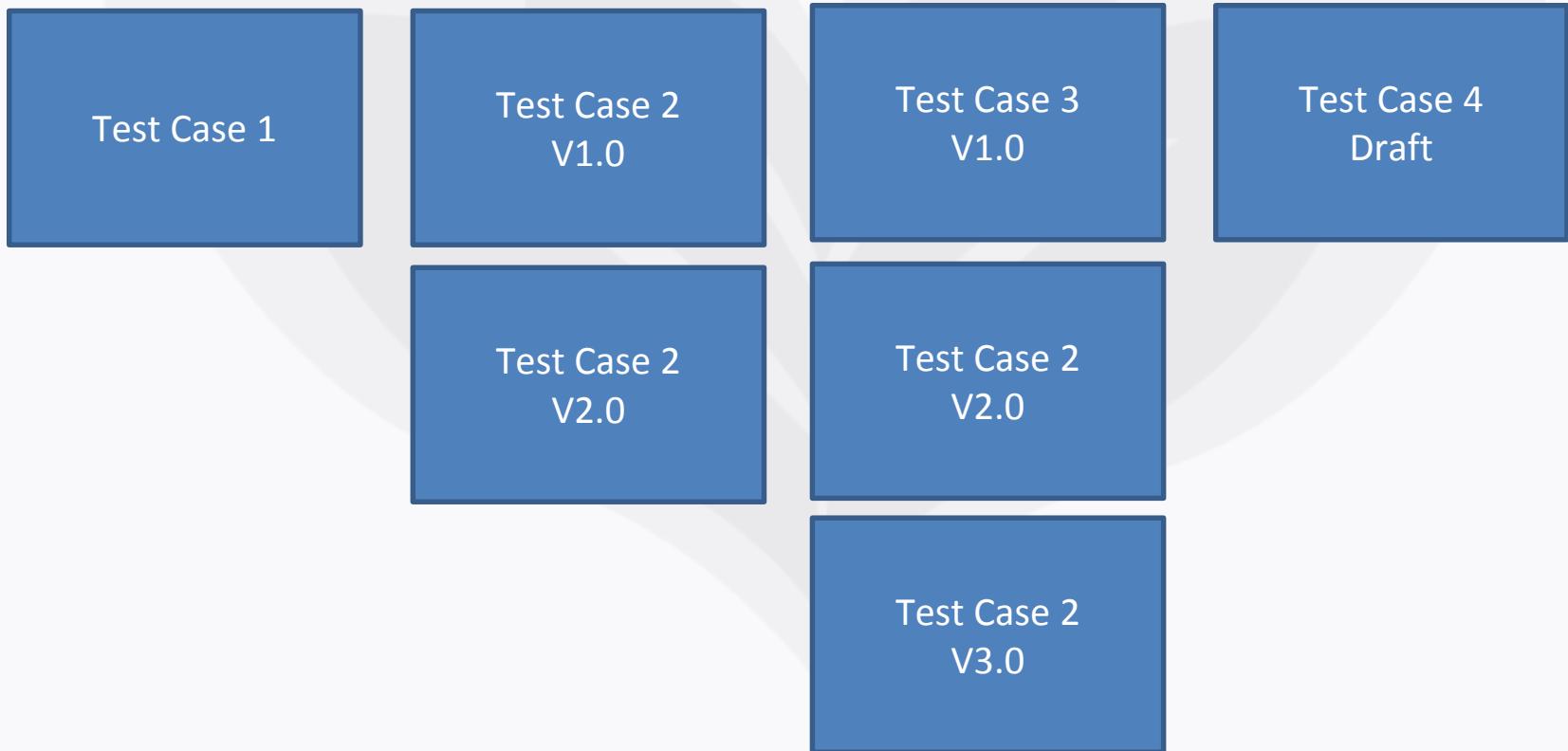


Test plan life cycle



FreeNest

Test Case Pool and version control



Free**Nest**

Test Management

- Using Testlink
- How to write test case?
- Creating a test plan

TestLink is a web based Test Management tool. The application provides Test specification, Test plans and execution, Reporting, Requirements specification and collaborate with well-known bug trackers.



Free**Nest**

Execution of Test Case

Test Engineer daily job ?

Different Roles in Testing



Test Designer
TestDesigner



Acceptance
Test Engineer
Acceptance
TestEngineer



Project Manager
ProjectManager



Test Manager
TestManager



System
Test Engineer
System
TestEngineer



Test Automation
Engineer
TestAutomation
Engineer



Developer
Developer

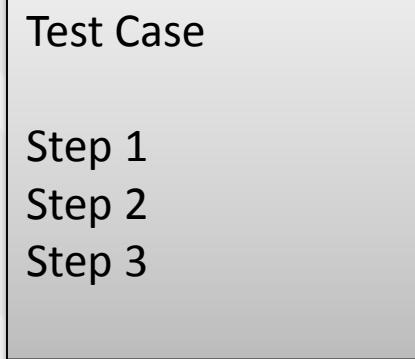
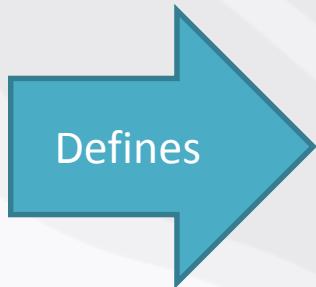
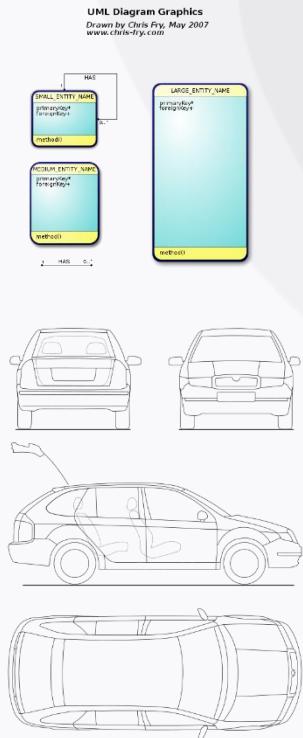


Free**Nest**



Test Designer ?

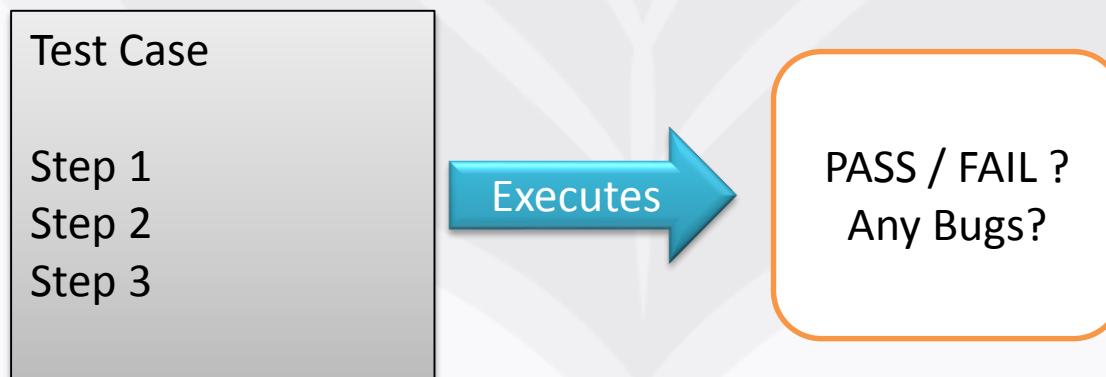
- Designer for test cases (has substance knowledge)





Tester ?

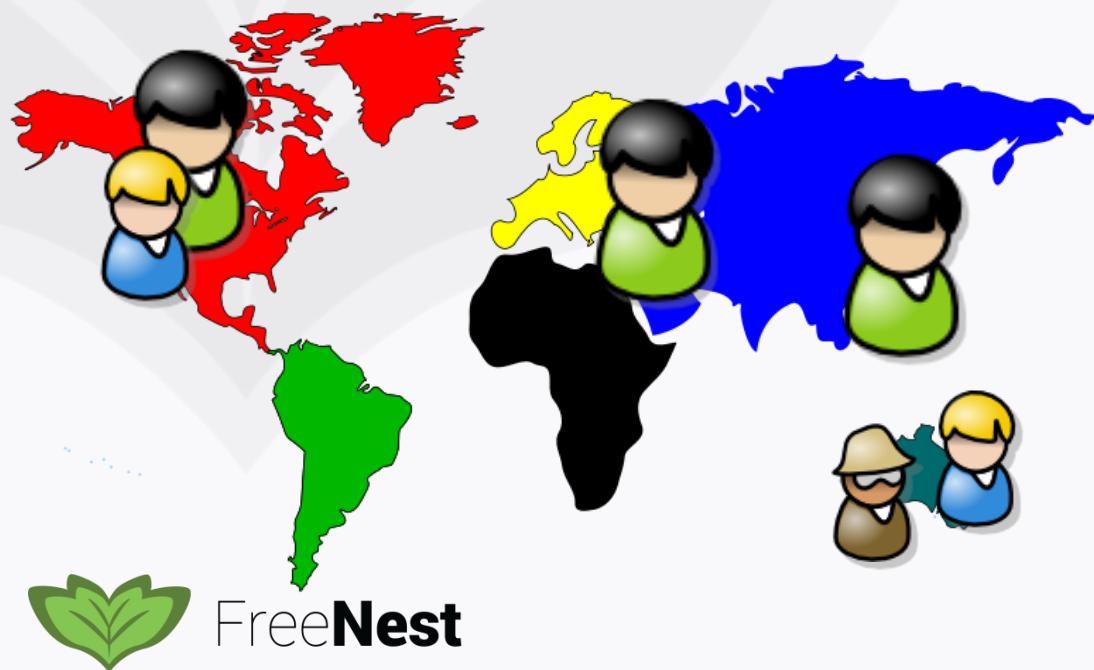
- Executes testing
- Creates reports and defect/bug/incident reports





Test Manager ?

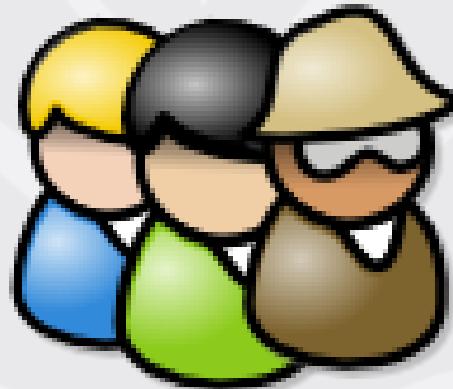
- Has control over testing process and resources
- This role exists in large projects and product lines
- Testing Strategy
- Management
- Master Test Planning



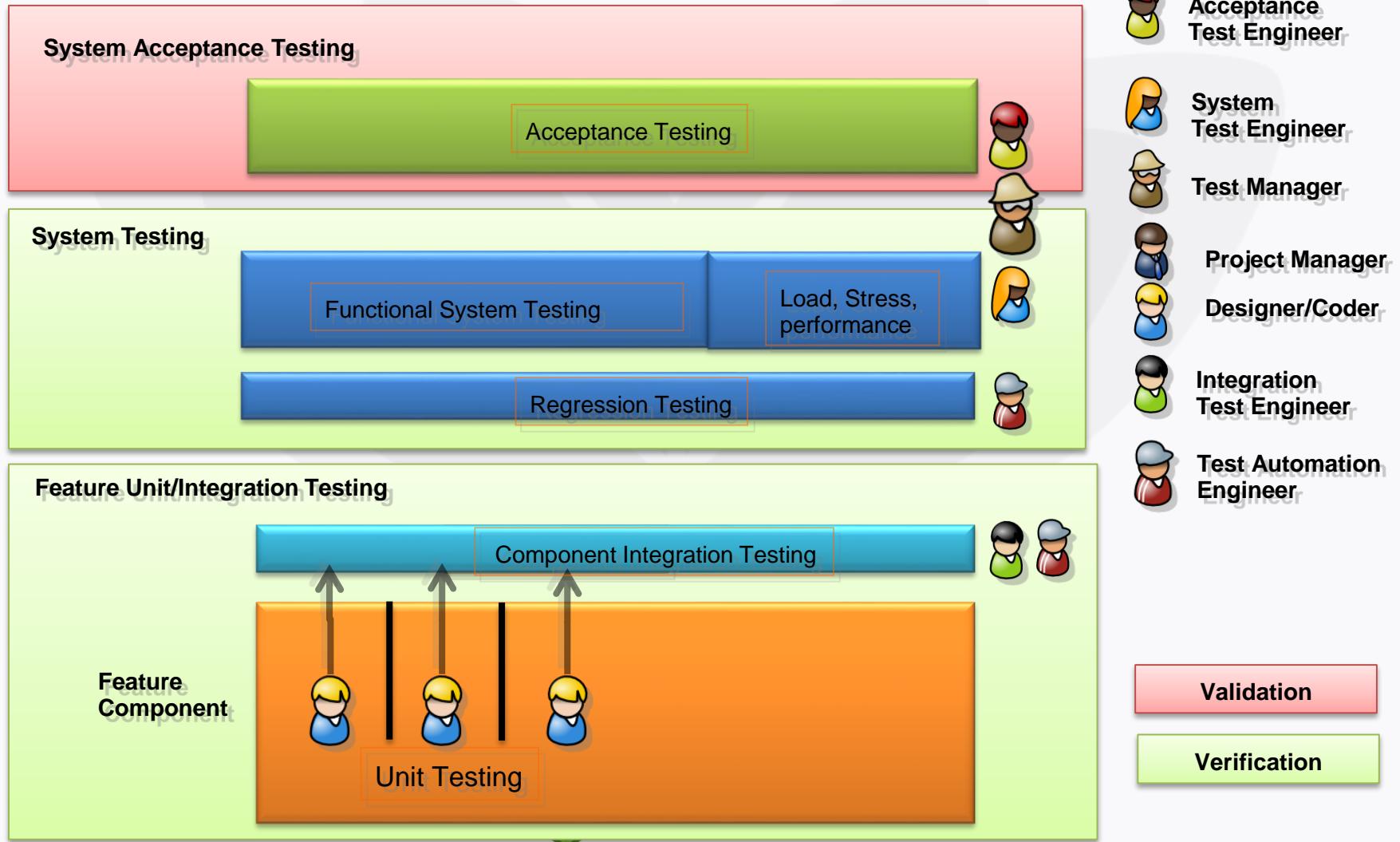
Test Management

- Reporting
- Executed Test case
- Not executed
- Blocked
- Who / Where / What?
- Founded bugs

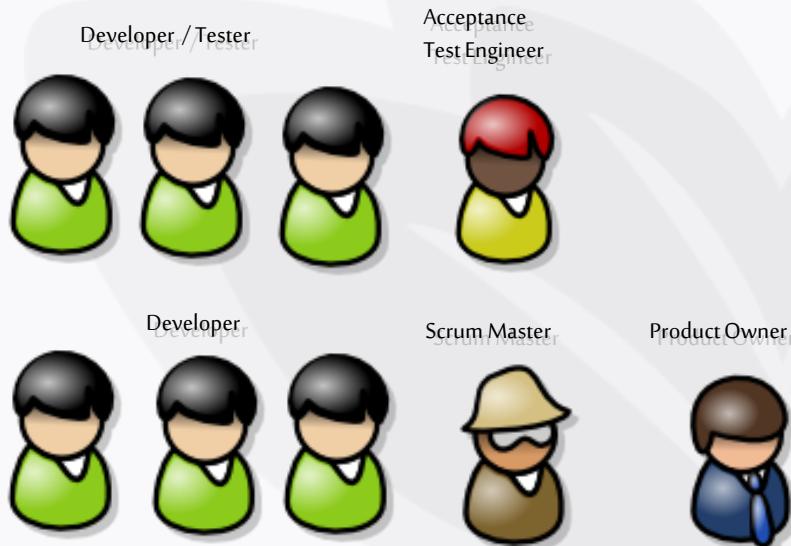
Test Engineer ?



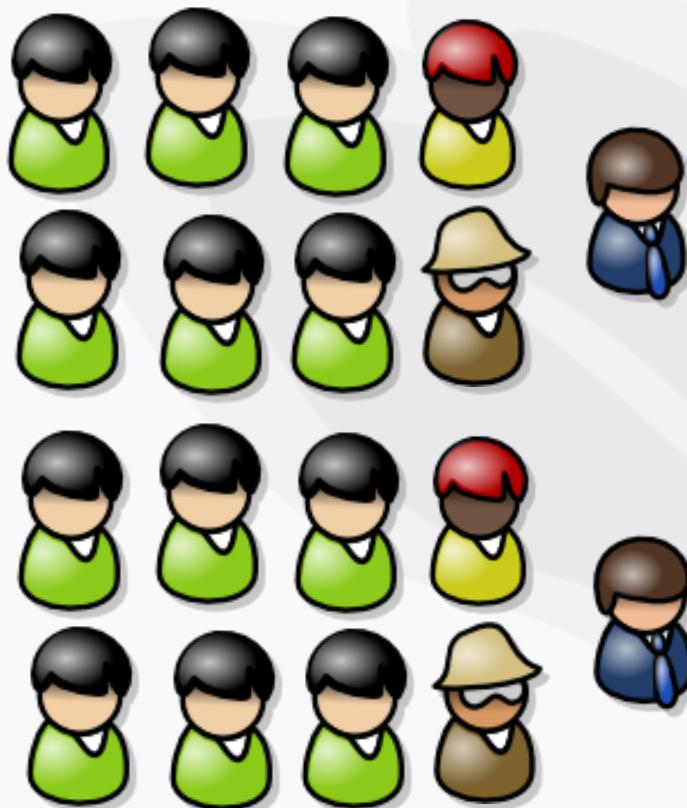
Example Product verification & validation organization



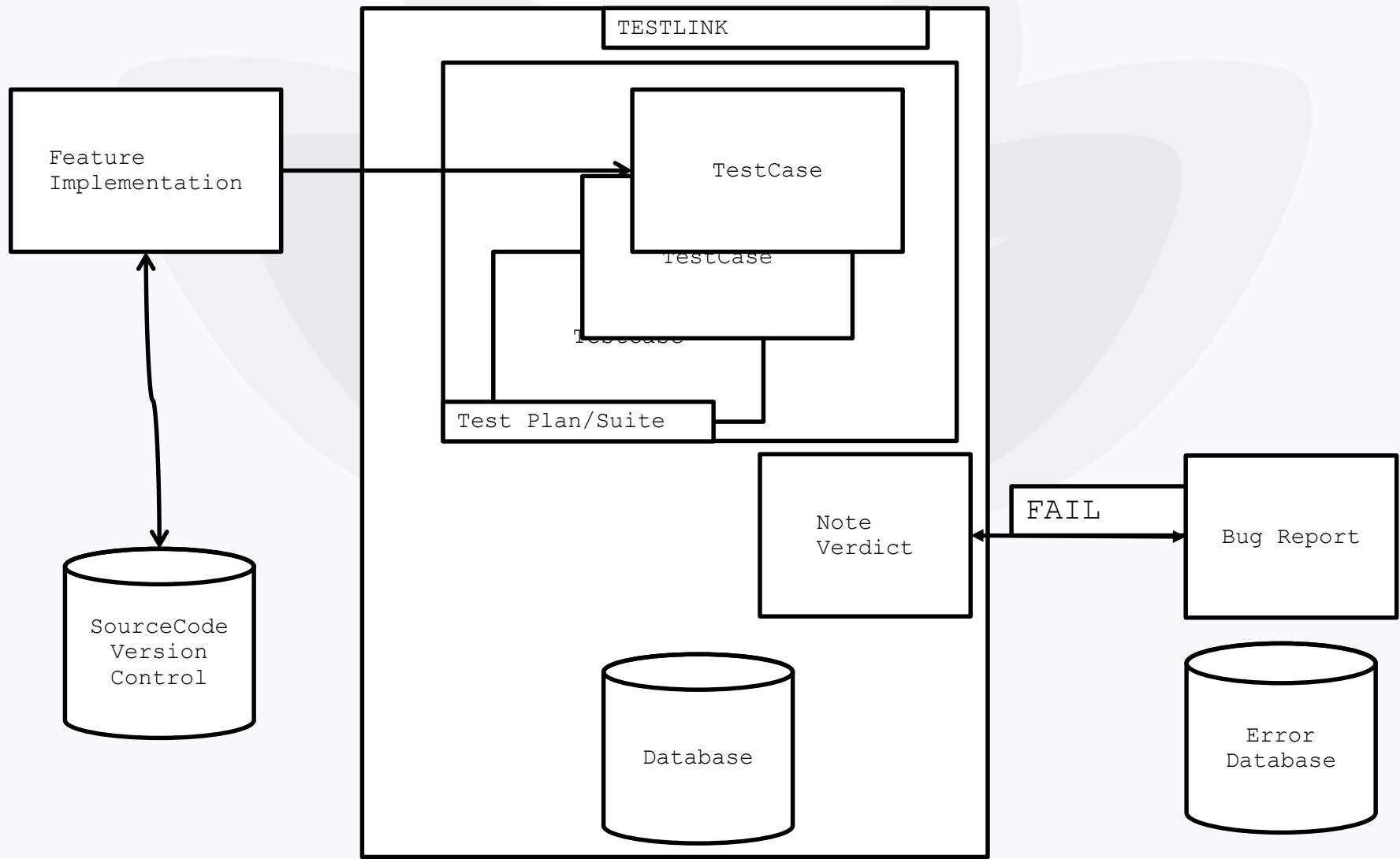
Agile Team and testing



Large Agile organization

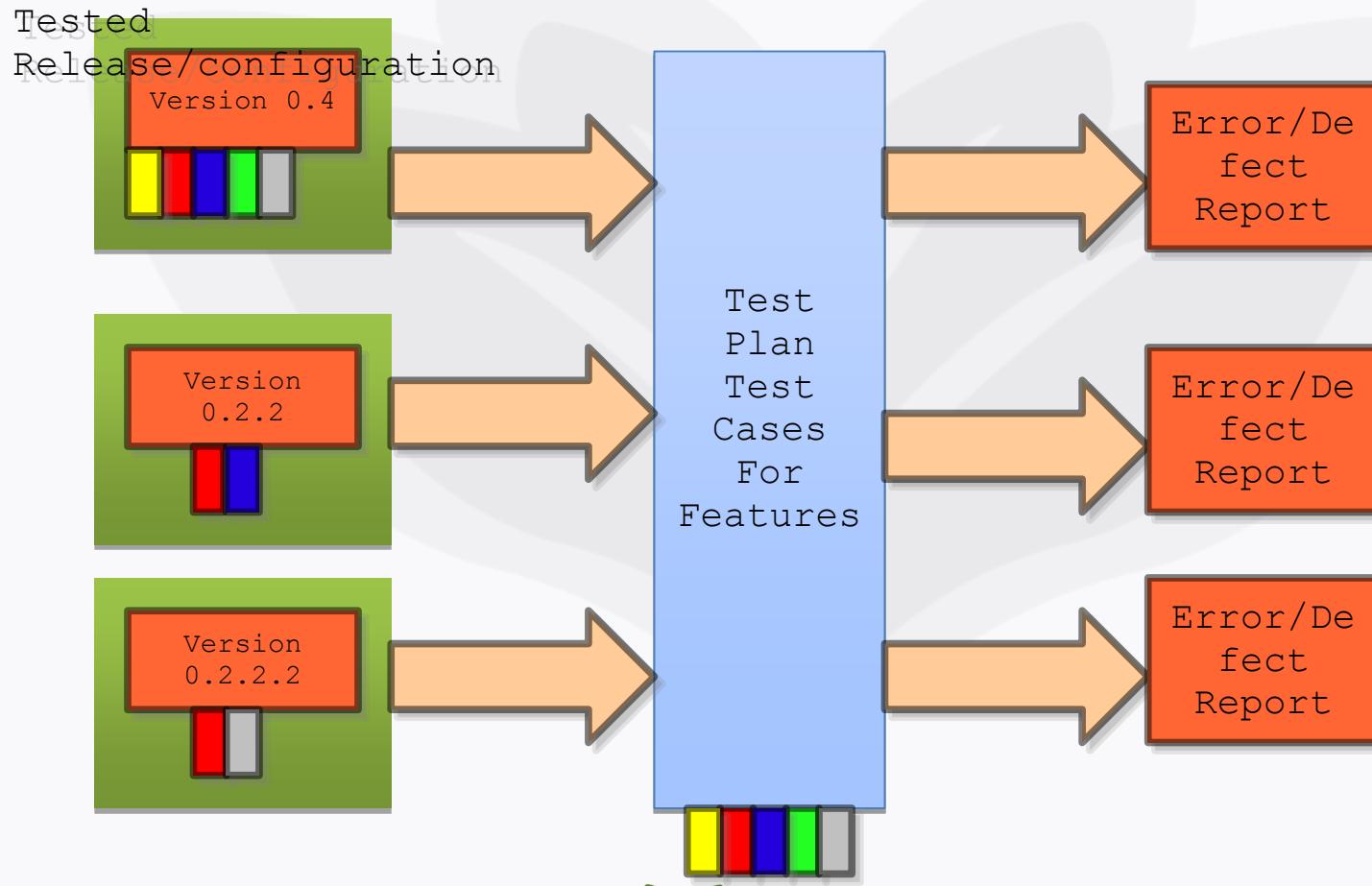


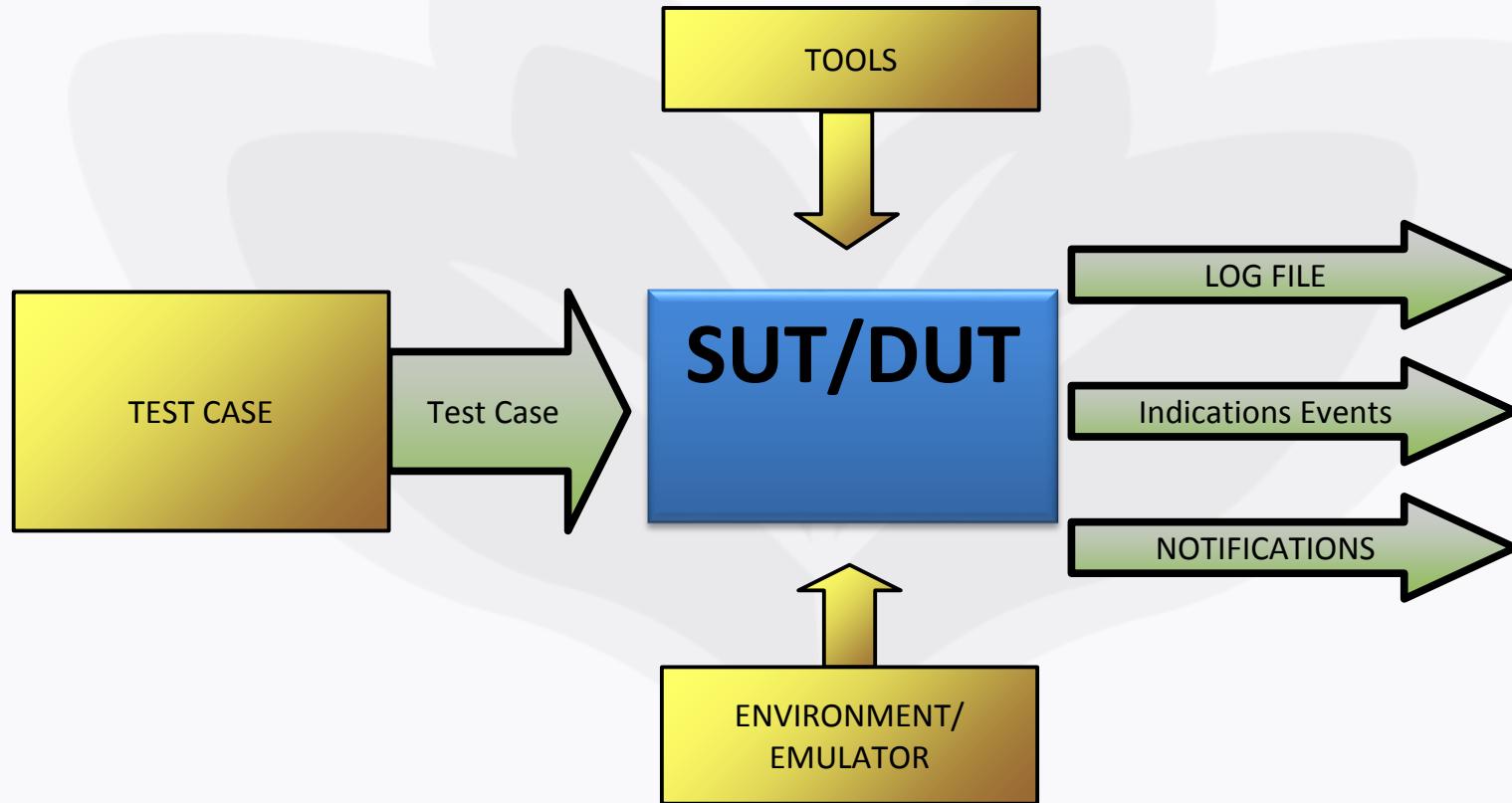
Testlink in brief



- What we can do with Testlink Tool

Selecting Configuration





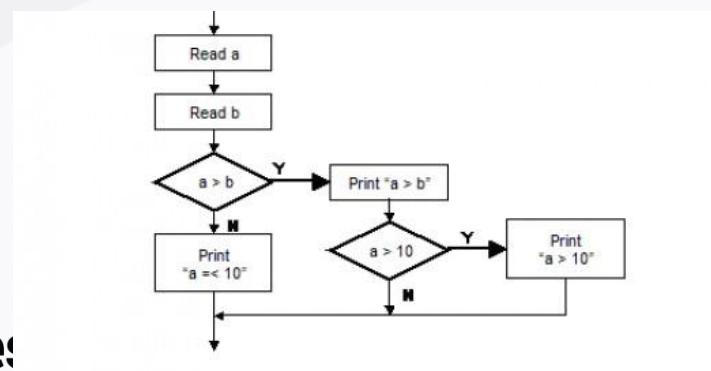
SUT = System Under Test
DUT = Device Under Test

Agile methods and testing

- We take branding seriously, The FreeNest Brand is our pride and joy.
- It is the message we want to send to the developer teams, customers and stakeholders.
- These guidelines must be followed in all FreeNest's print and electronic communications, marketing and informing.
- Documentation defines logo guidelines, typography and the use of colors.

Test Automation

- We take branding seriously, The FreeNest Brand is our pride and joy.
- It is the message we want to send to the developer teams, customers and stakeholders.
- These guidelines must be followed in all FreeNest's print and electronic communications, marketing and informing.
- Documentation defines logo guidelines, typography and the use of colors.



Test Automation Frameworks

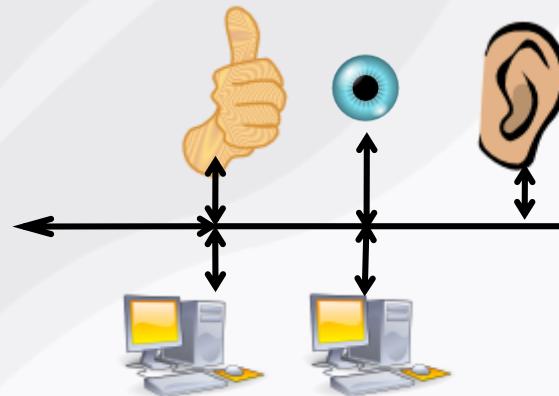


Why Test Automation Framework?

- There has to be common way to communicate with test environment. Test Automation Framework provides it



Test Team communicates with common language



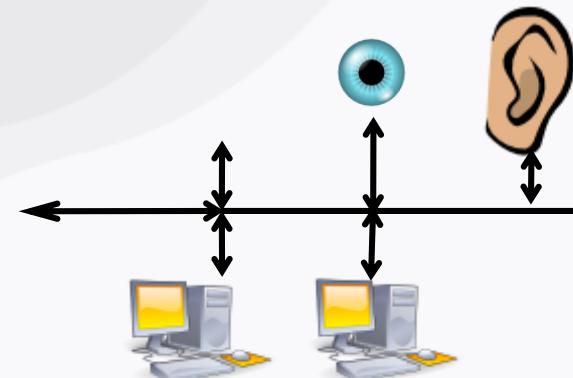
Test Automation Framework provides a way to Control testing environment with common way



Free**Nest**

STAF as foundation for test automation

- The Software Testing Automation Framework (STAF) is an open source, multi-platform, multi-language framework designed around the idea of reusable components, called services
- STAF removes the tedium of building an automation infrastructure, thus enabling you to focus on building your automation solution
- The STAF framework provides the foundation upon which to build higher level solutions, and provides a pluggable approach supported across a large variety of platforms and languages.

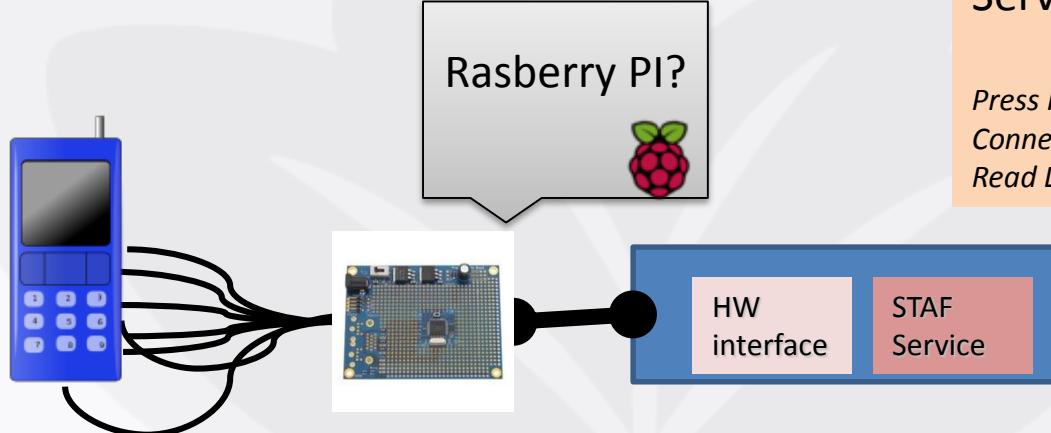


<http://staf.sourceforge.net/>

Test Environment Example

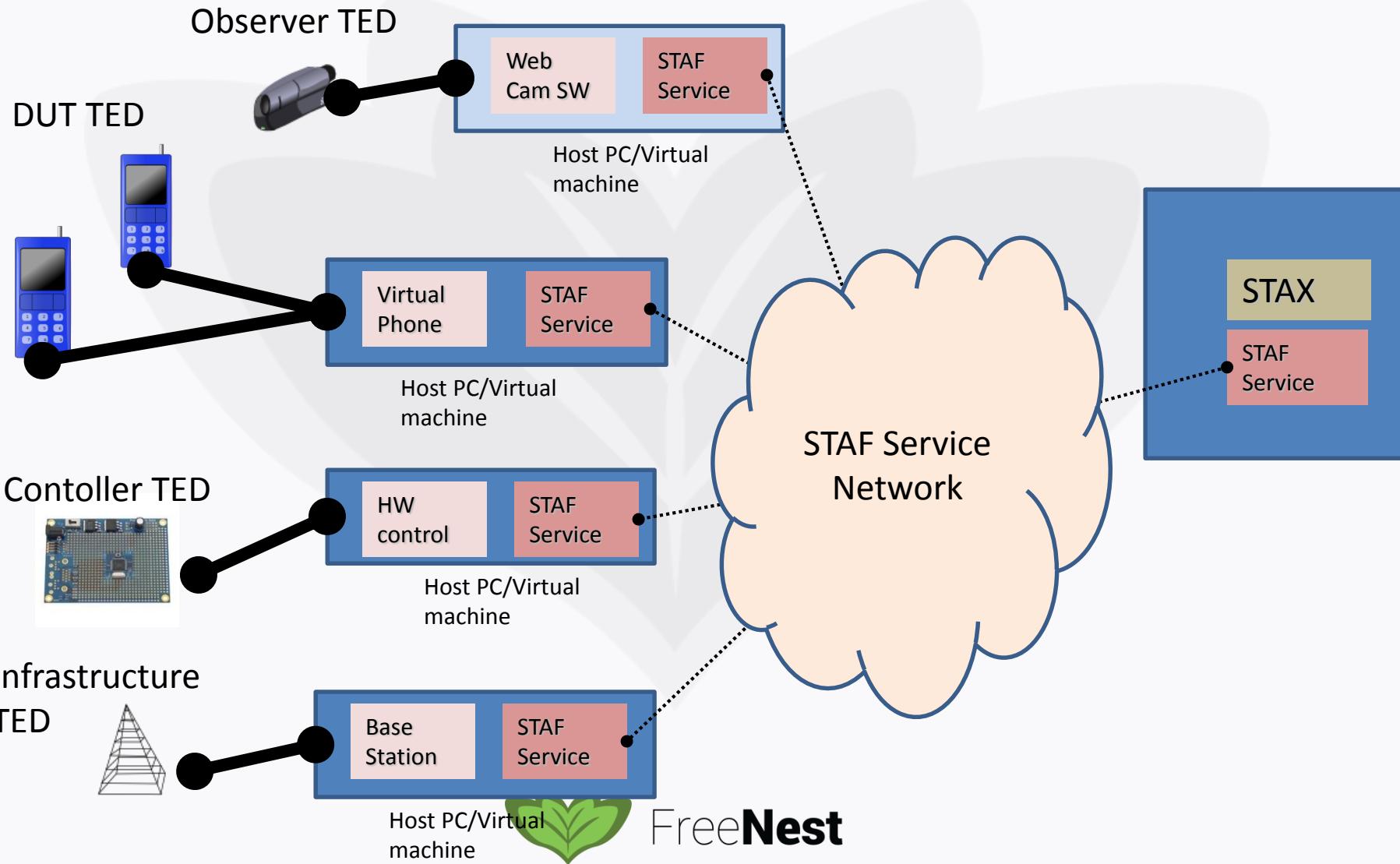
- TED = Test Environment Device
- Test Environment Devices are "lego" bricks of for test scenario building.
- TED's can be used as service interfaces to DUT (Design/Device under test)
- TED service has to be programmed to use DUT API/interface
- Test Automation needs support for virtual "eyes" and "arms". TED's can be also used such a way
- All TED 's are communicating using STAF Network
-

TED "virtual hand" 😊

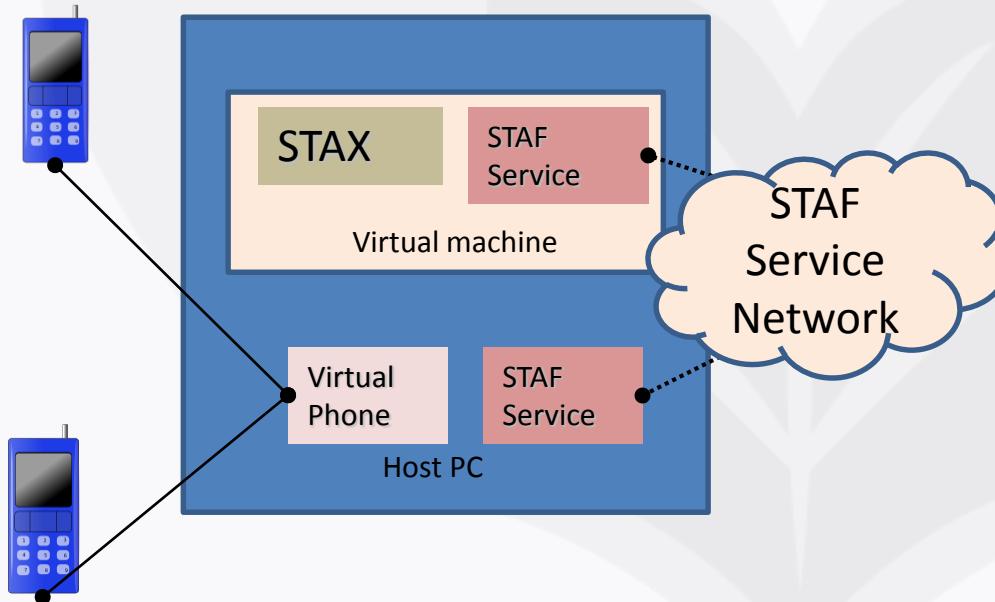


<http://www.raspberrypi.org/>

Using STAF/STAX in PMR testing



DUT TED



```
#Presettings  
Case_Passed=False  
  
PMRRadioA.Select DMO_Mode  
PMRRadioB.Select DMO_Mode  
PMRRadioA.Select DMO_Channel( 10)  
PMRRadioB.Select DMO_Channel (10)  
#Case start  
PMRRadioB ResetDMOCallCounter  
PMRRadioB CountDMOCall  
  
Do while call_count < 100  
{  
    PMRRadioB MakeDMOCall(5, 5)  
    call_count ++  
}  
  
If (PMRRadioB .CallCounter = 100)  
{  
    Case_Passed= True  
}  
  
# Case End
```

- STAX executes test script using PMRRadio virtual phone service interface
- Virtual phone implementation is DUT specific

Ideas?

- There is need for hardware to control keypad with machine
- It's just a one small part of whole test automation
- How to control screen? You will need another test tool?

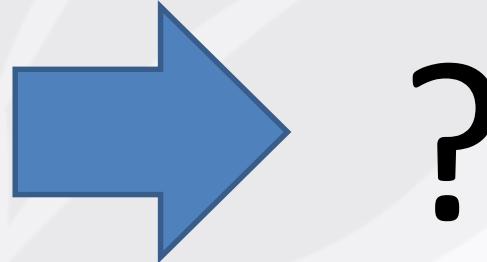
Code Analyzing

Code Complexity

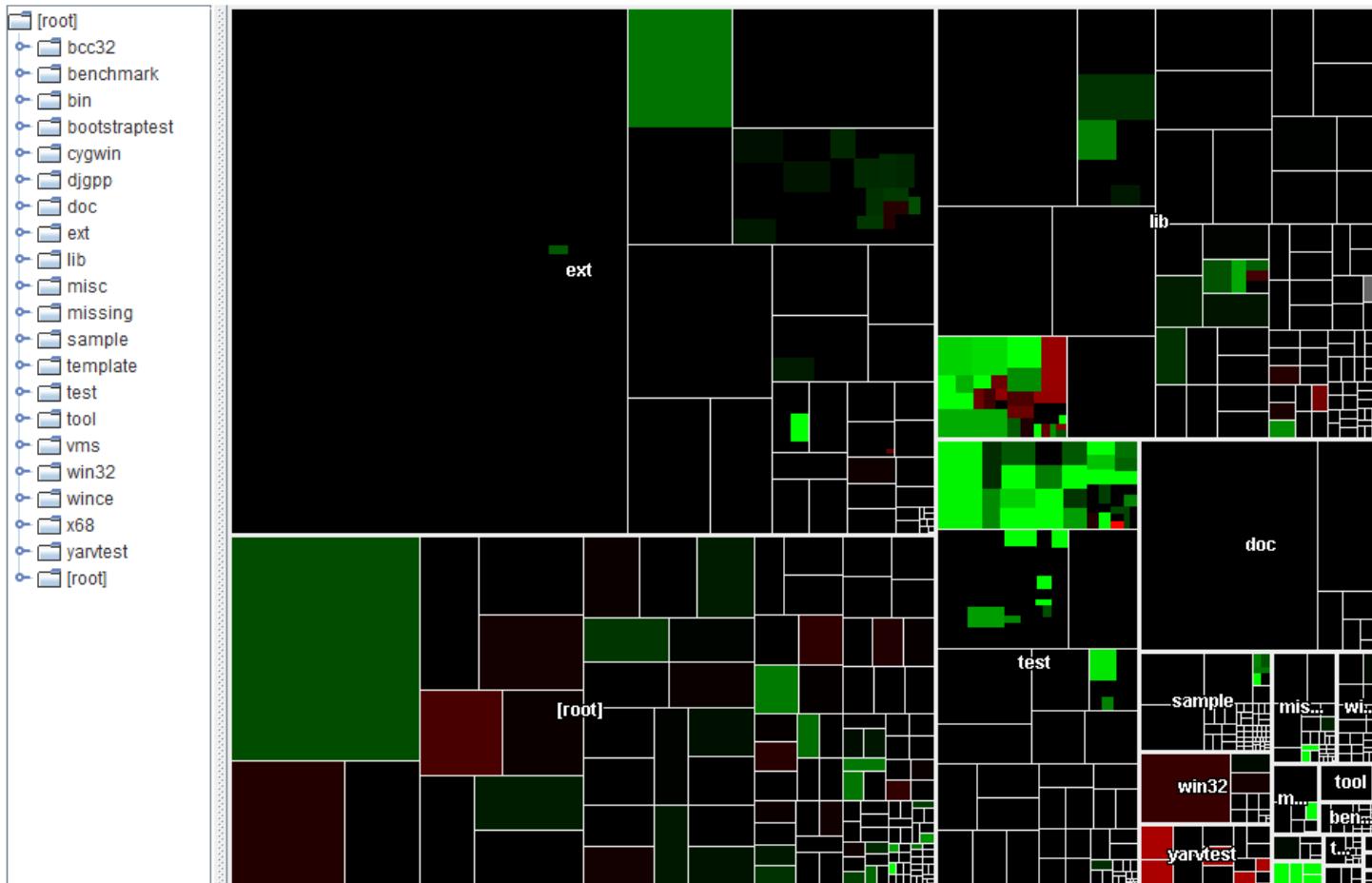
- Example tool CCCC
- <https://wiki.jenkins-ci.org/display/JENKINS/CCCC+Plugin>
- <http://sourceforge.net/projects/codeanalyze-gpl/?source=recommended>

Static Code Analyze

```
10 CLS: PRINT TAB(15); "Polysons": PRINT  
20 PI=3. 14159: TP=PI*2  
30 INPUT "Number of Sides";SD  
40 INPUT "Size (5 to 80)";SZ  
50 SP=TP/SD  
60 SCREEN 1  
70 FOR A=0 TO TP+SP/2 STEP SP  
90 X=SZ*SIN(A)+100  
100 Y=SZ OS(A)+96  
110 IF A=0 THEN 130  
120 LINE (X0, Y0)-(X, Y)  
130 X0=X:Y0=Y  
140 NEXT  
150 GOTO 150
```

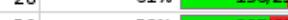
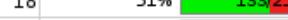
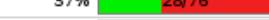
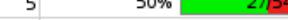
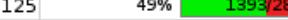
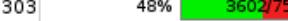
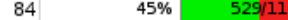
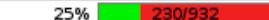
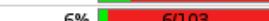
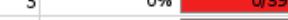


Static Code Analyze – Heat Map



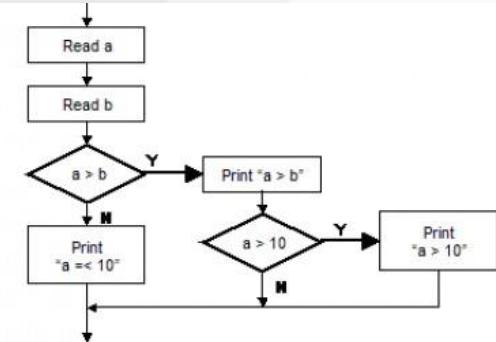
<http://www.statsvn.org/demo/ruby/>

Code Coverage

Package	# Classes	Line Coverage	Branch Coverage	Complexity
hudson.tools	6	89% 	81% 	0
hudson.views	15	75% 	N/A 	0
hudson.search	28	67% 	60% 	1.421
hudson.scheduler	7	67% 	40% 	2.294
hudson.node_monitors	28	61% 	39% 	0
hudson.slaves	50	52% 	37% 	1.6
hudson.model.listeners	5	52% 	46% 	1.667
hudson.triggers	18	51% 	37% 	1.5
hudson.widgets	5	50% 	17% 	1
hudson.matrix	27	50% 	32% 	1
hudson	125	49% 	43% 	2.645
hudson.model	303	48% 	39% 	1.46
hudson.tasks	68	46% 	32% 	1
hudson.security	84	45% 	28% 	1.724
hudson.tasks.junit	17	43% 	36% 	3
hudson.maven	79	43% 	32% 	1.406
hudson.util	167	35% 	29% 	1.832
hudson.scm	83	35% 	25% 	3
hudson.util.spring	10	34% 	24% 	1.238
hudson.diagnosis	3	31% 	7% 	0
hudson.maven.reporters	37	29% 	19% 	2.182
hudson.tasks.test	15	19% 	6% 	0
hudson.scm.browsers	20	8% 	2% 	2
hudson.logging	6	8% 	4% 	0
hudson.os.solaris	10	6% 	5% 	0
hudson.os.windows	4	3% 	0% 	0
hudson.lifecycle	11	2% 	2% 	0
hudson.util.ina	20	0% 	0% 	1
hudson.tasks.labelers	3	0% 	0% 	0
hudson.org.apache.tools.ant.taskdefs.csvlib	8	0% 	0% 	1.667
hudson.org.apache.tools.ant.taskdefs	1	0% 	0% 	0
hudson.fsp	5	0% 	0% 	0

Code Coverage – Branch Coverage

The percentage of branches that have been exercised by a test suite. 100% branch coverage implies both 100% decision coverage and 100% statement coverage.



<http://booster911.hubpages.com/hub/BranchDecisionTesting>

Code Analyze – Cyclomatic Complexity

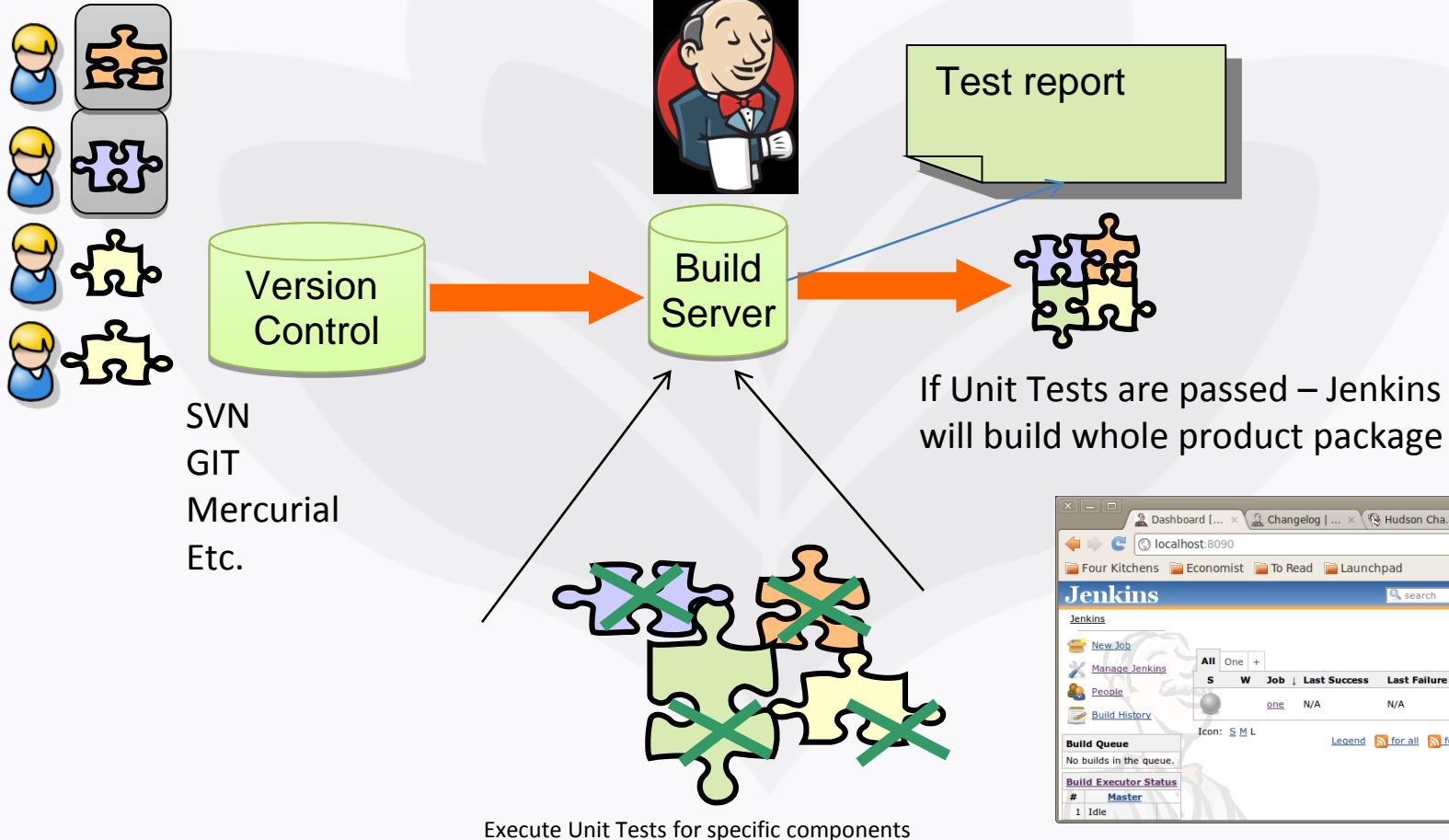
?

http://en.wikipedia.org/wiki/Cyclomatic_complexity

Build server ?

- <http://buildbot.twistedmatrix.com/waterfall>

Jenkins as build server

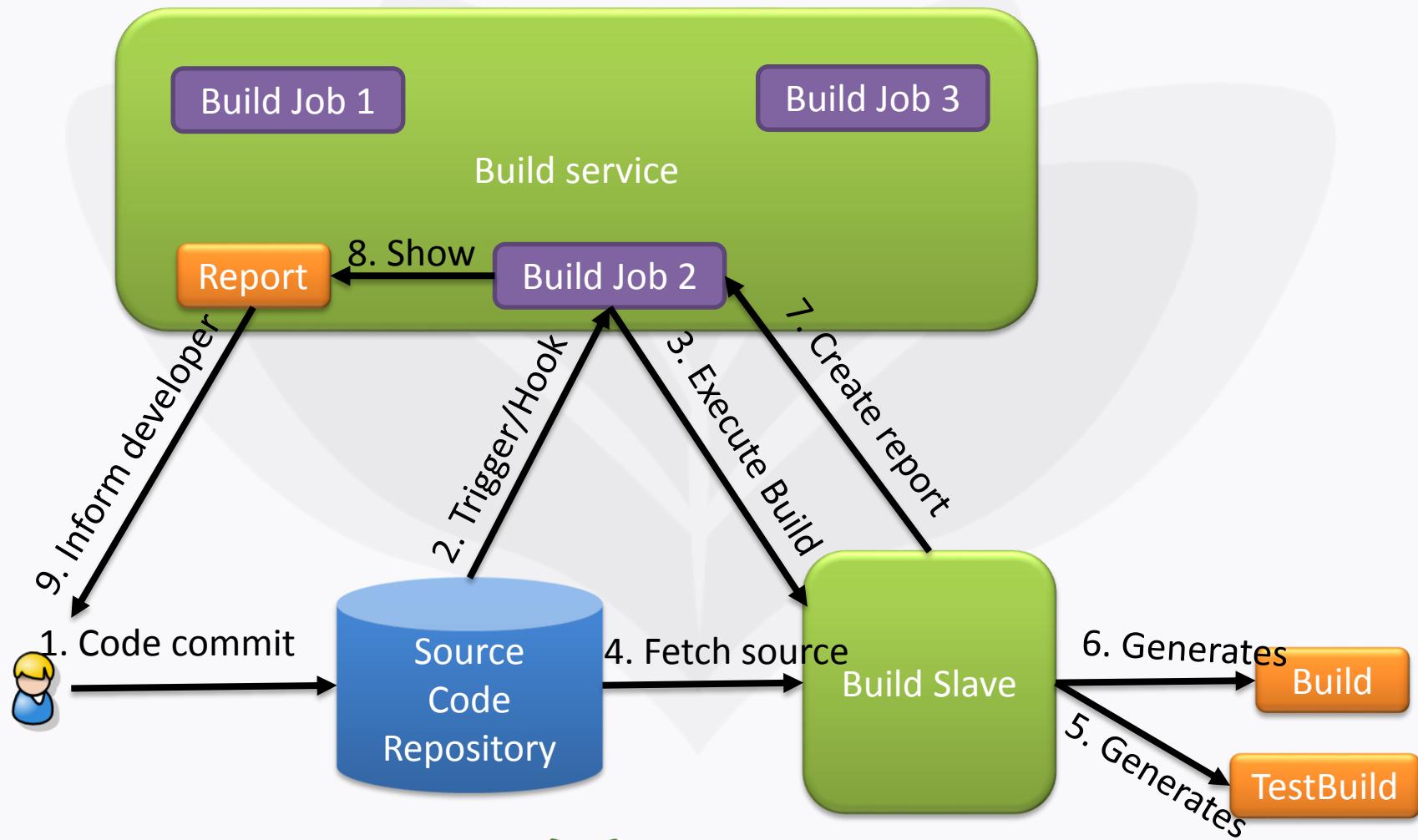


<http://www.youtube.com/watch?v=Ago7hpp6xZI&feature=related>

Youtube Jenkins CI

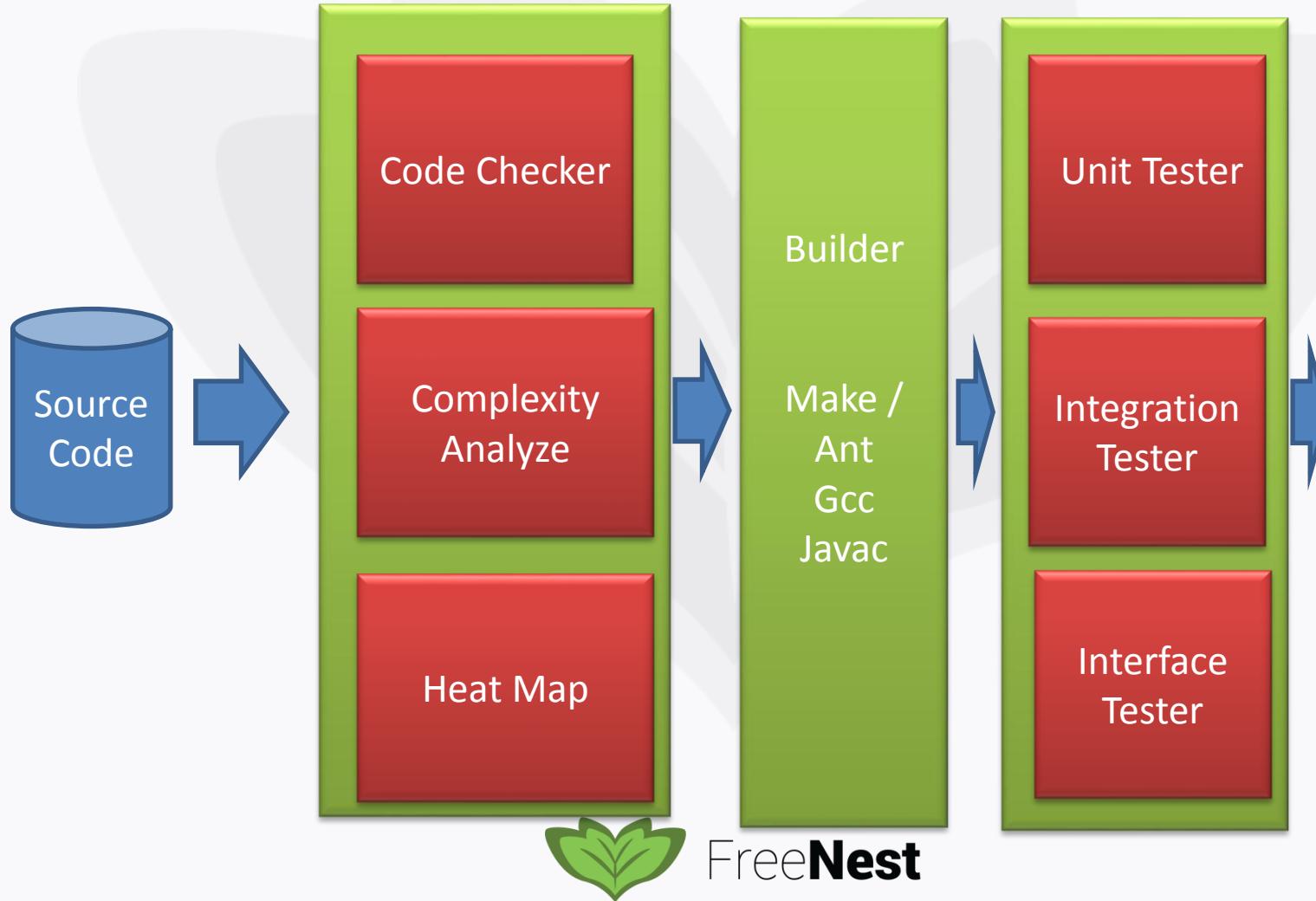


Simple continuous integration



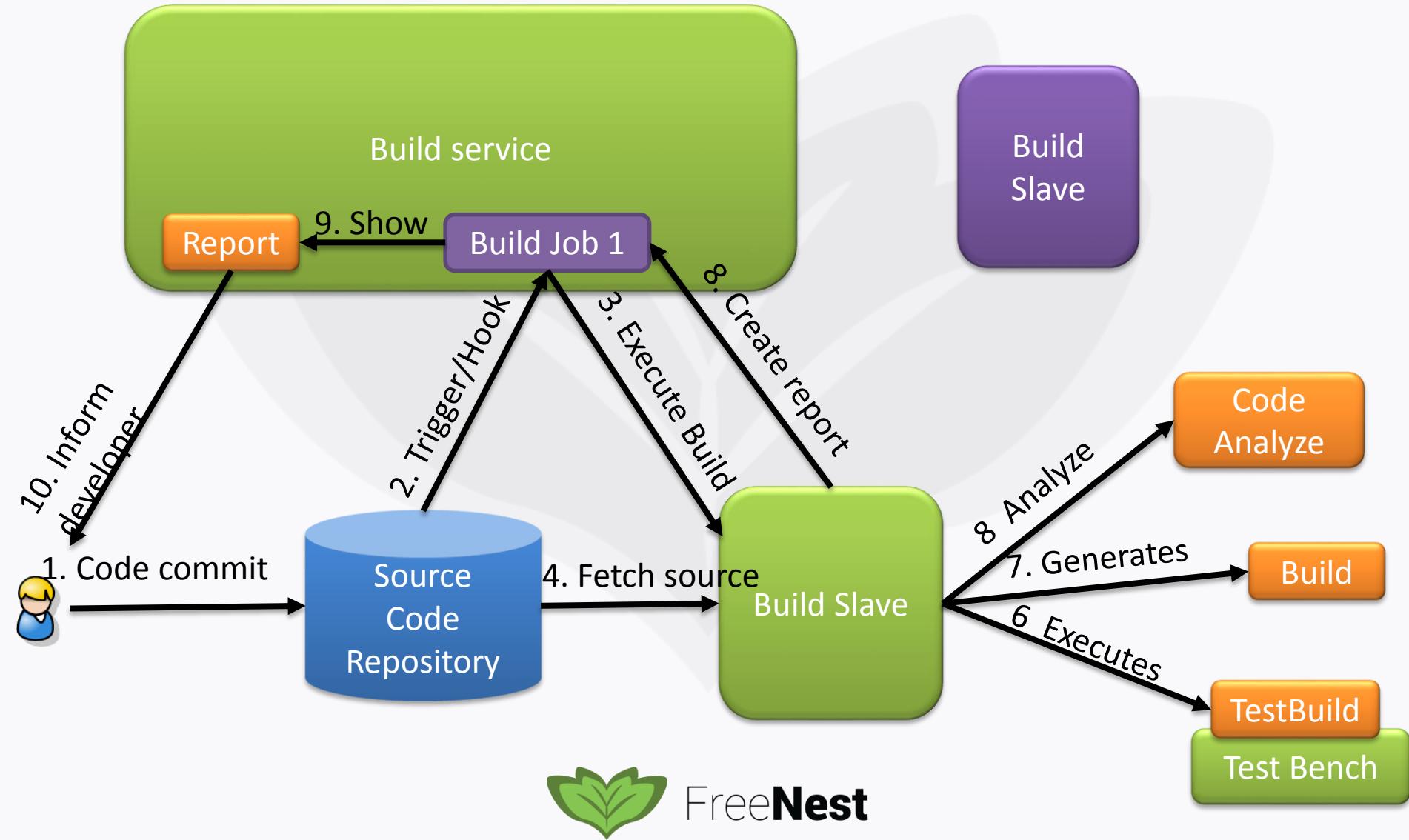
FreeNest

Continuous Integration and code analyze?



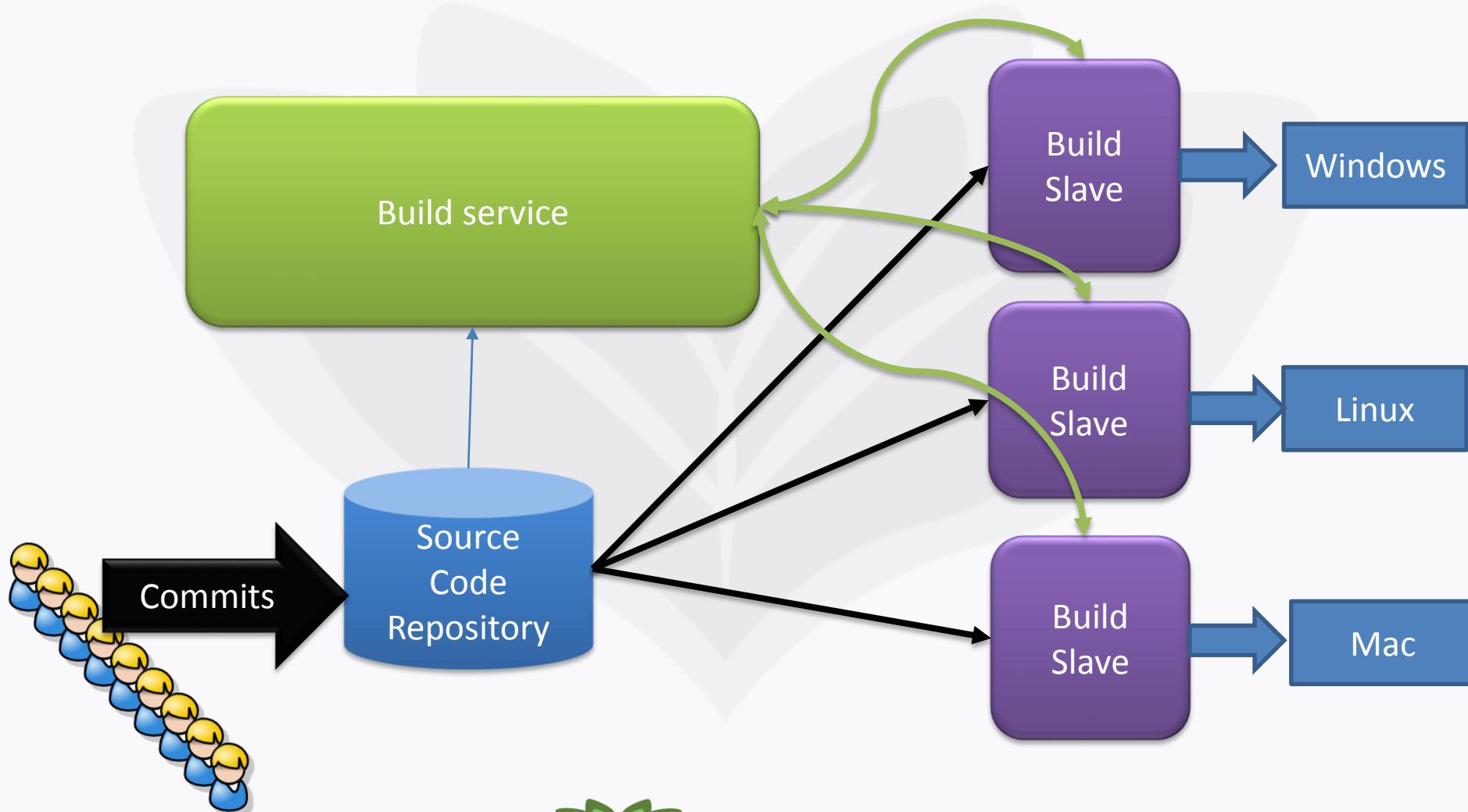
Free**Nest**

Advanced Continuous Integration

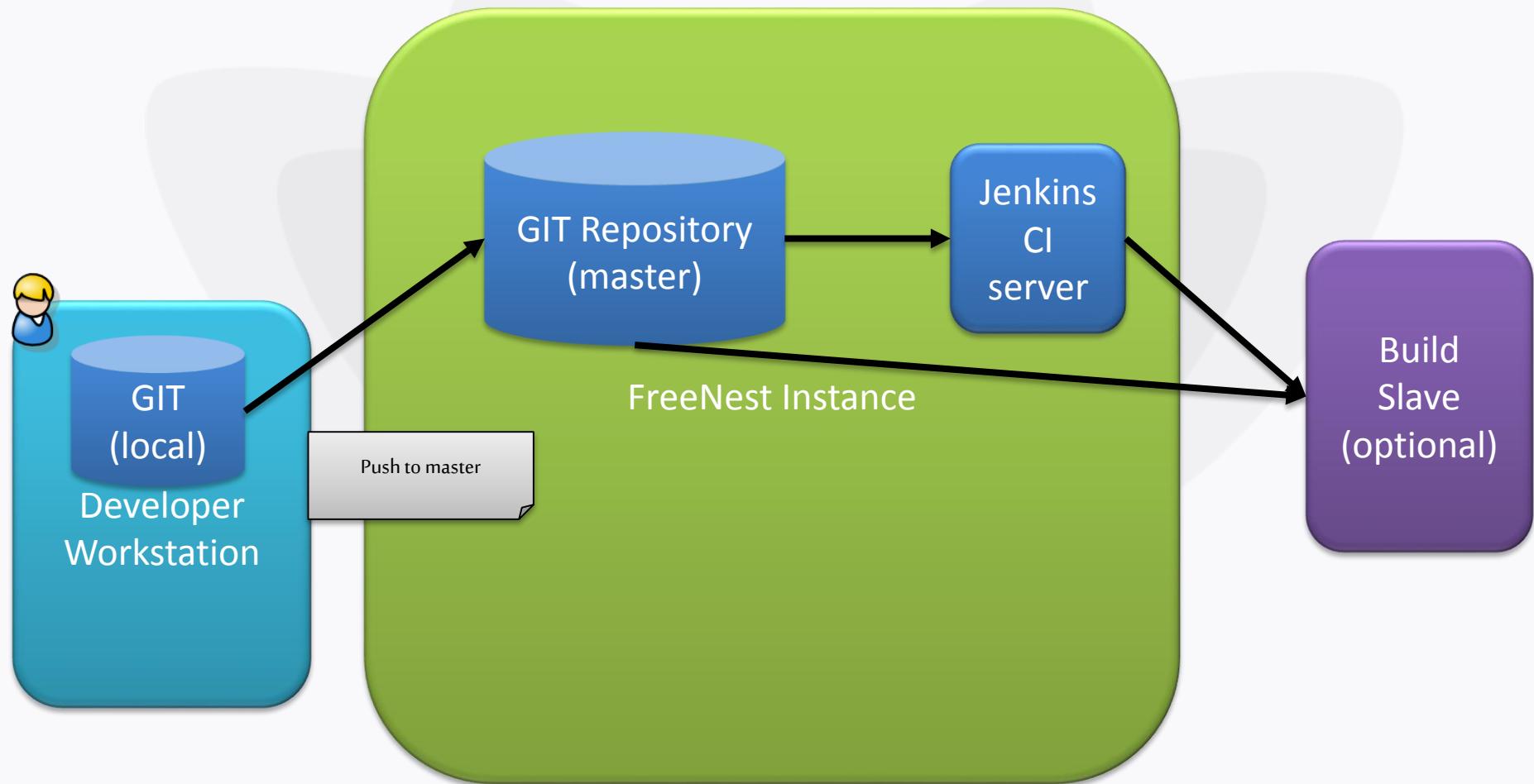


Free**Nest**

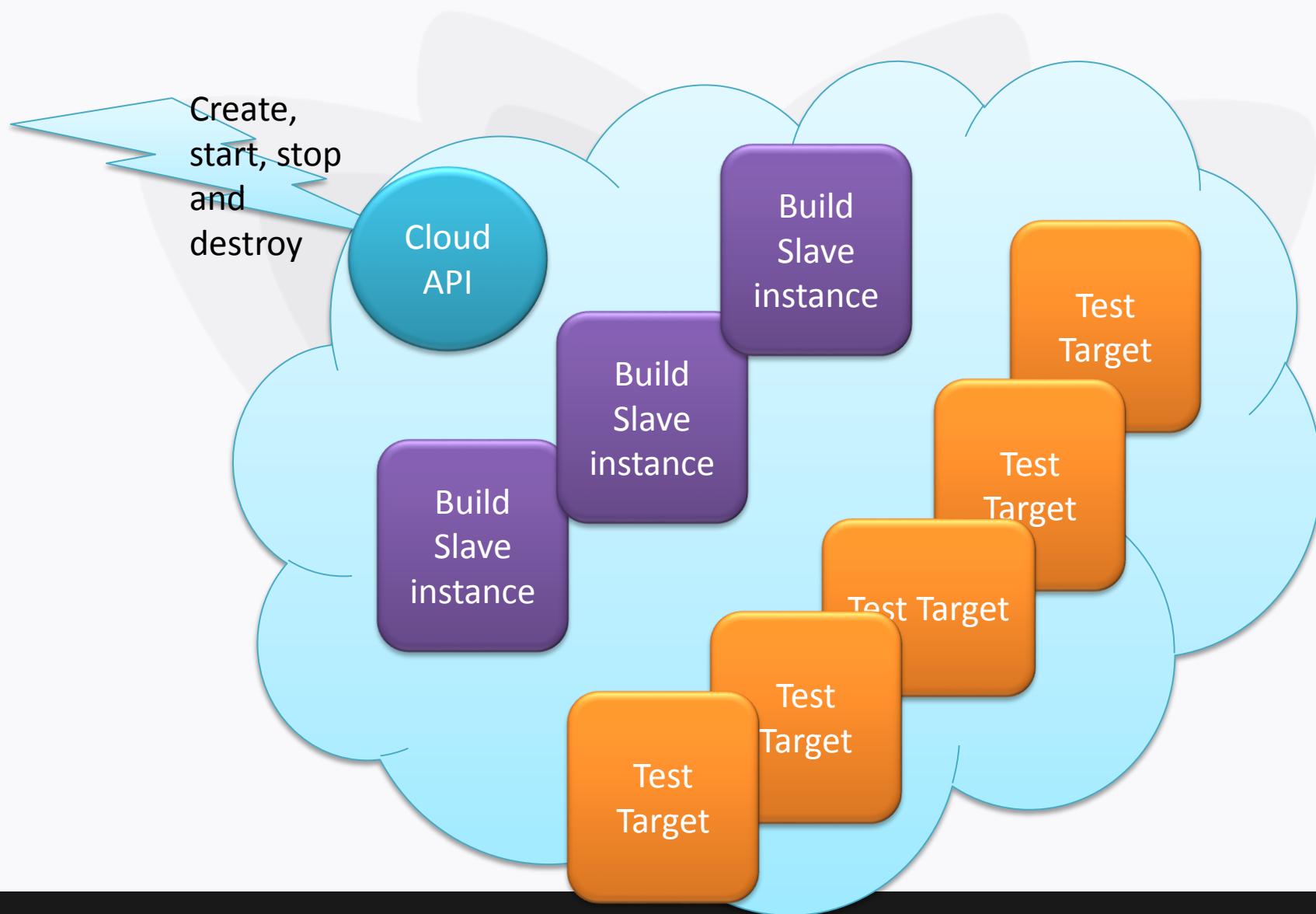
Multiple release targets



Build Service and FreeNest



Cloud as scalable CI environment



Error Management



Free**Nest**

Concepts

Failure – Fault – Defect - Bug -
Incident – Failure - Error

Example forum thread:

<http://www.allinterview.com/showanswers/36257.html>

ISTQB syllabus



Free**Nest**

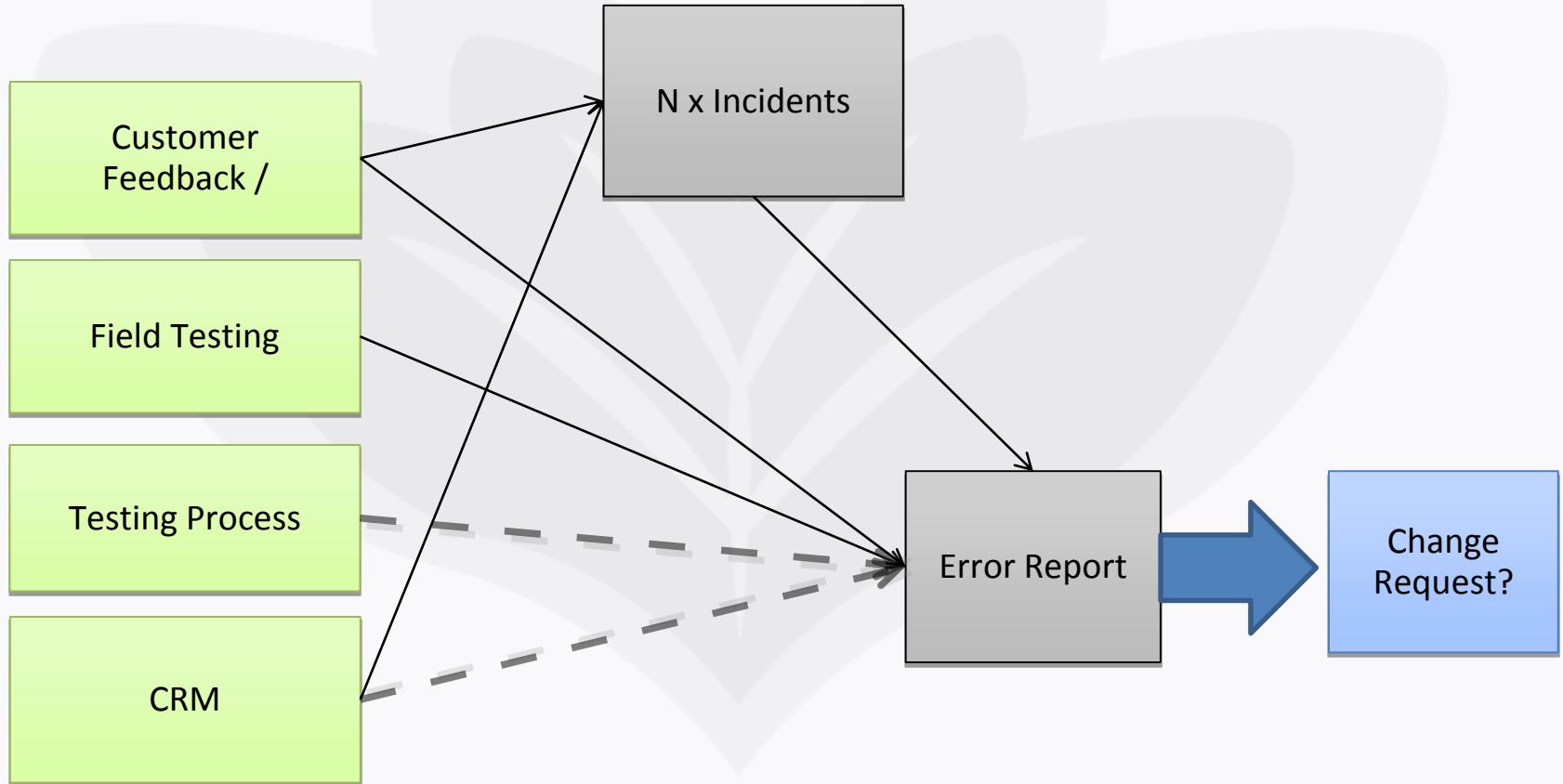
Error/Bug/Defect Report

- .Defect/Bug/Error ID
- .Reporter
- .Time
- .Founded where
- .Which way?
- .Test Case
- .Test Setup/Configuration
- .Describe scenario?
- .Attachments? Picture/Log/etc..

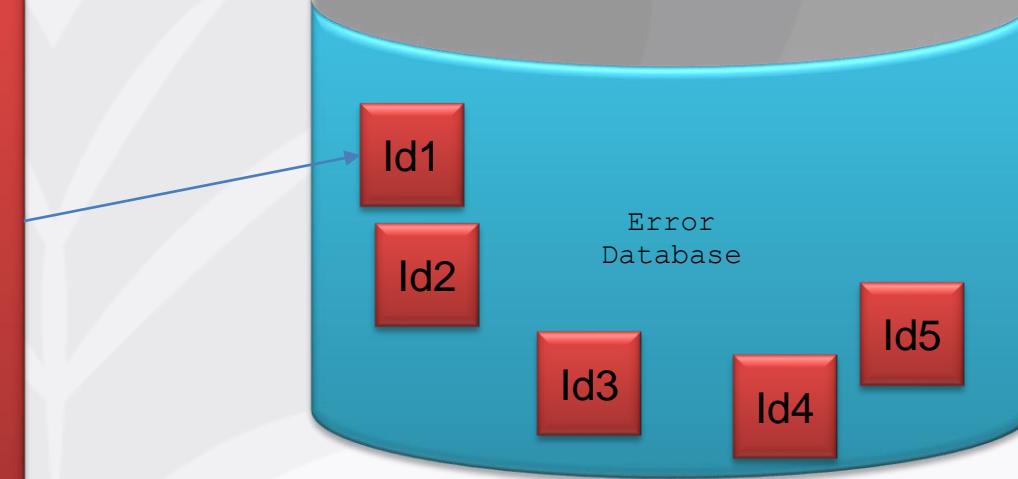


Free**Nest**

Sources for error report



- .Defect/Bug/Error ID
- .Reporter
- .Time
- .Founded where
- .Which way?
- .Test Case
- .Test Setup/Configuration
- .Describe scenario?
- .Attachments? Picture/Log/etc..



Nature of bug

Location?

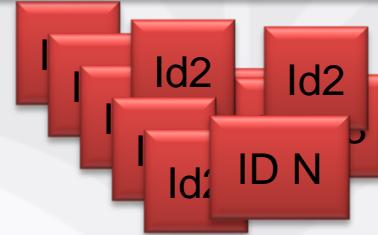
Where is the bug is found?

Priority?

Which one should fixed first?

Severity?

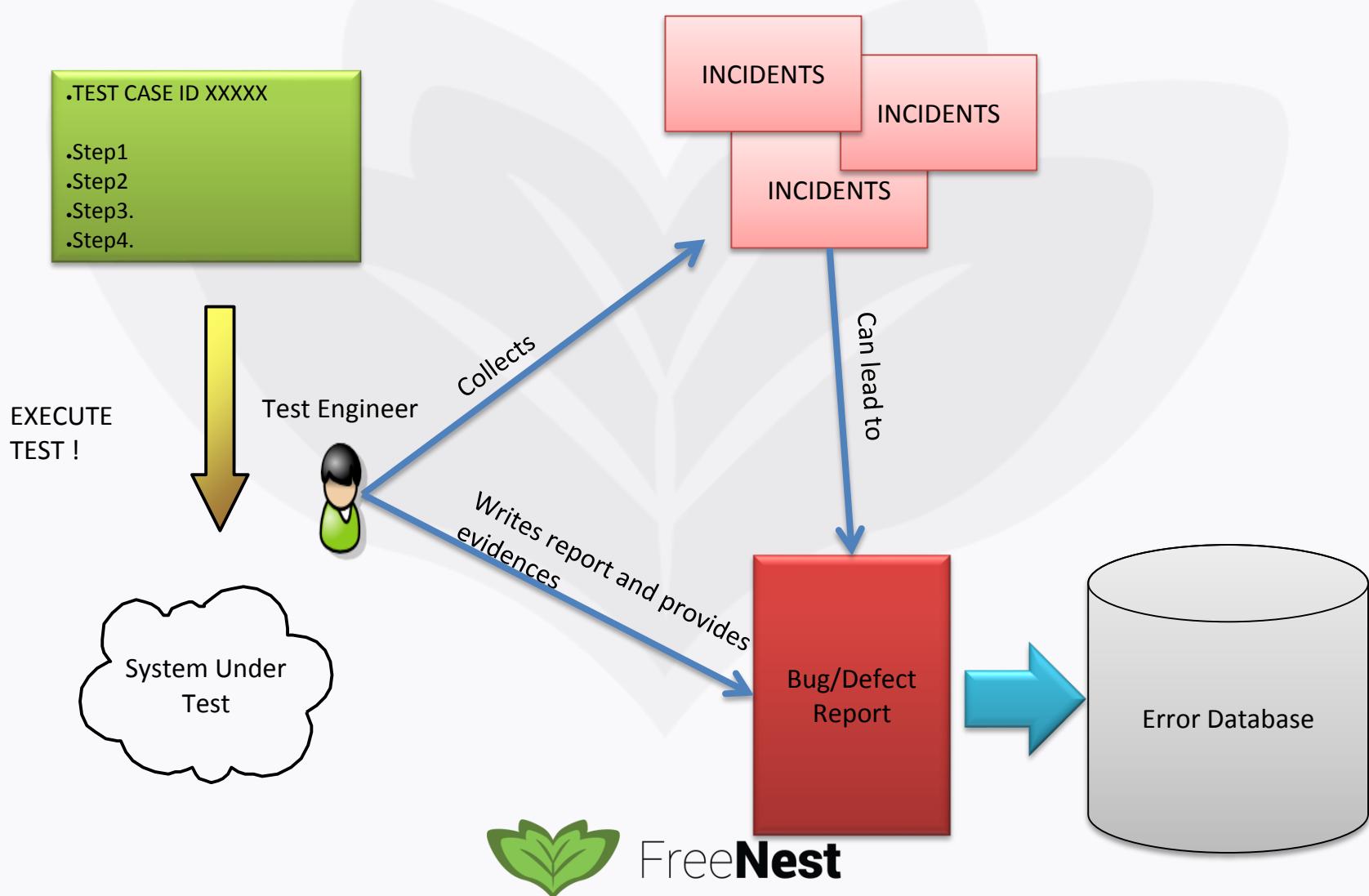
How fatal is the bug?



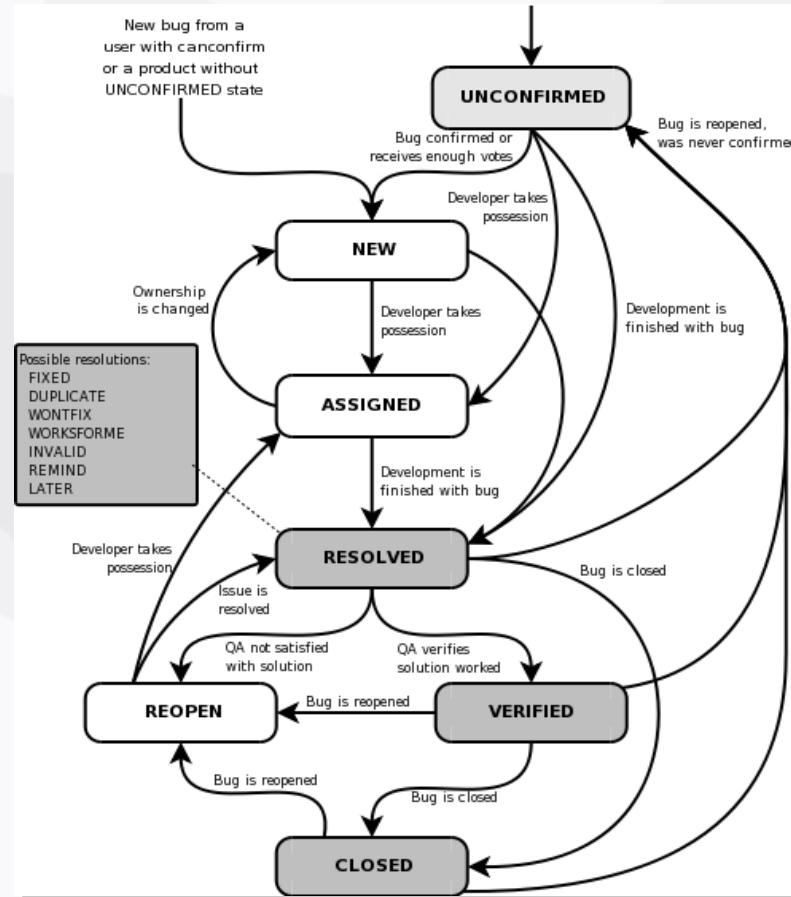
Target?

When the bug should be fixed and verified ?

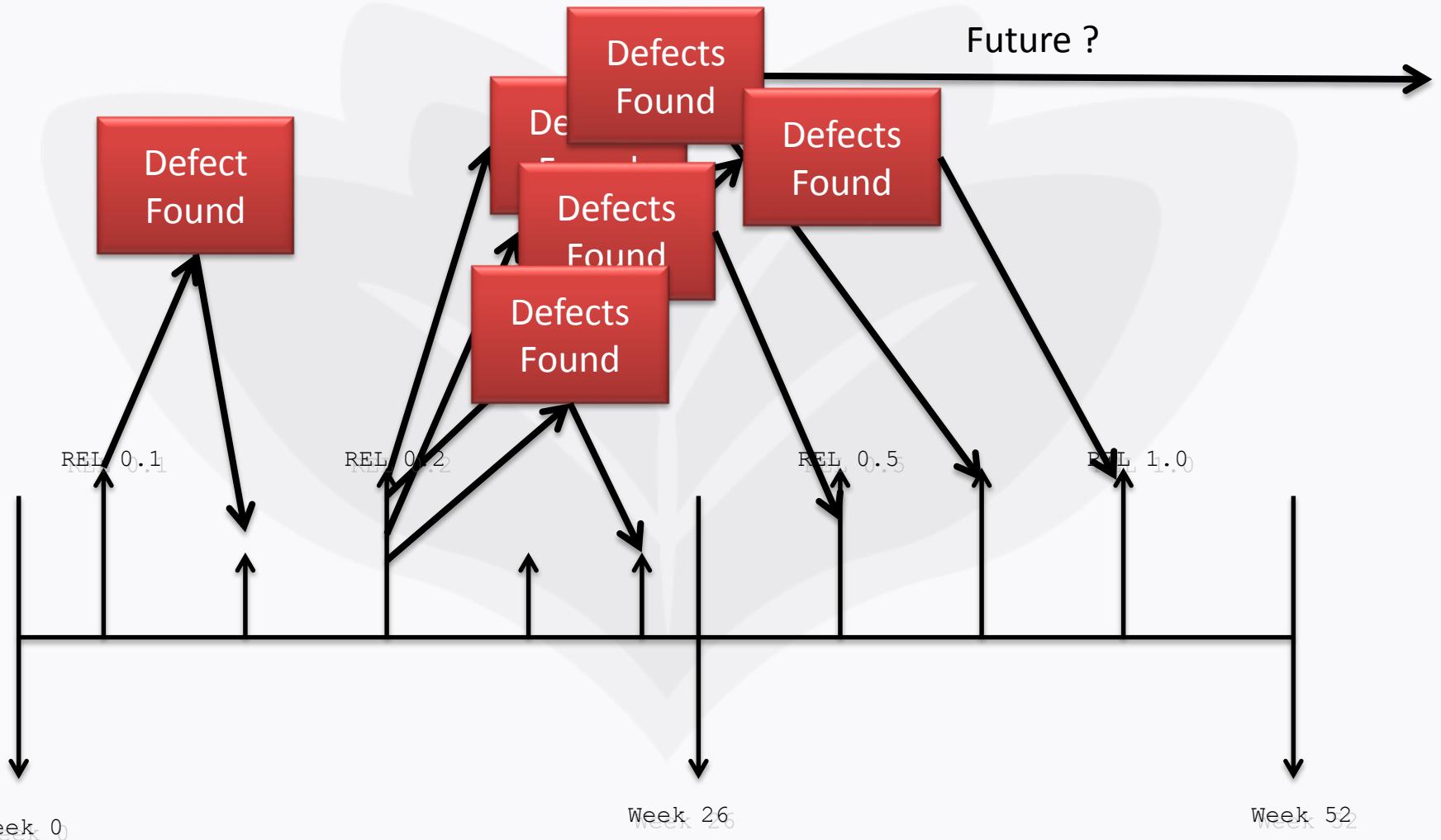
Test Case execution and error reporting?



Example of error process



FreeNest



FreeNest

Bugzilla as example

- What is Bugzilla?

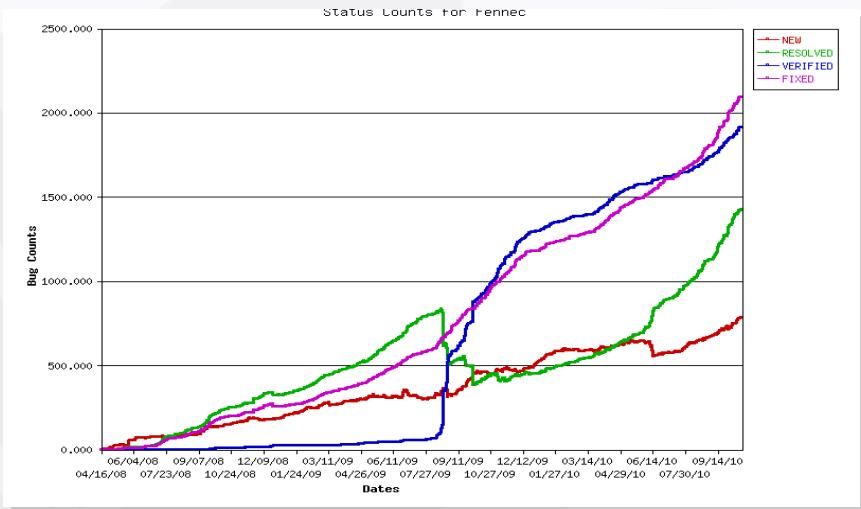
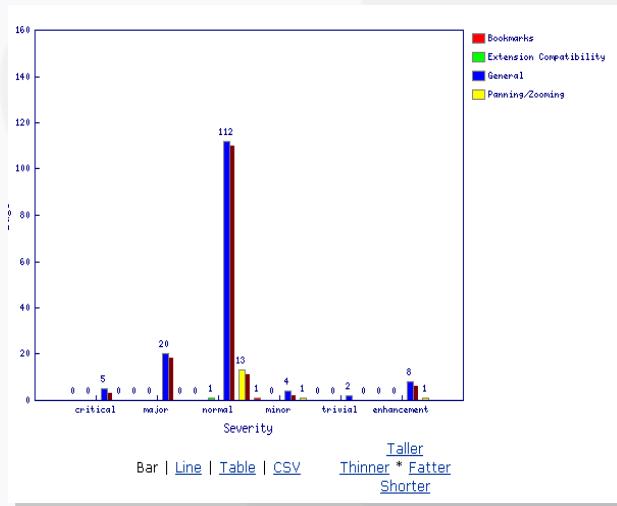
Bugzilla is a "Defect Tracking System" or "Bug-Tracking System". Defect Tracking Systems allow individual or groups of developers to keep track of outstanding bugs in their product effectively. Most commercial defect-tracking software vendors charge enormous licensing fees. Despite being "free", Bugzilla has many features its expensive counterparts lack. Consequently, Bugzilla has quickly become a favorite of thousands of organizations across the globe.



<http://www.bugzilla.org/>

<http://www.bugzilla.org/installation-list/>

Reporting and metrics



A table titled "Status Counts for Fennec" showing the count of bugs by component and severity. The Y-axis lists components: Bookmarks, Extension Compatibility, General, and Panning/Zooming. The X-axis lists severities: critical, major, normal, minor, trivial, enhancement, and Total. The data is identical to the bar chart above.

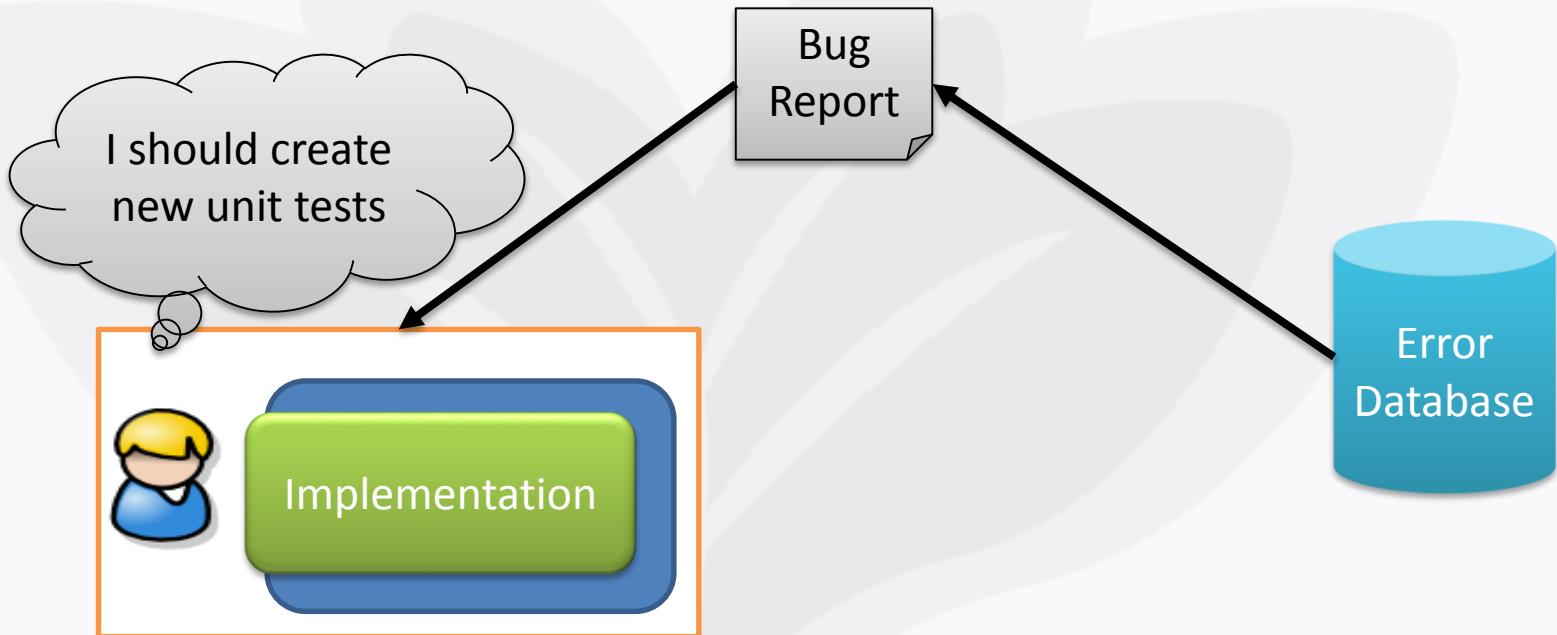
Component		critical	major	normal	minor	trivial	enhancement	Total
Bookmarks	.	.	.	1	.	.	1	
Extension Compatibility	.	.	1	.	.	.	1	
General	5	20	112	4	2	8	151	
Panning/Zooming	.	.	13	1	.	1	15	
Total	5	20	126	6	2	9	168	

Bar | Line | Table | CSV
Edit this report

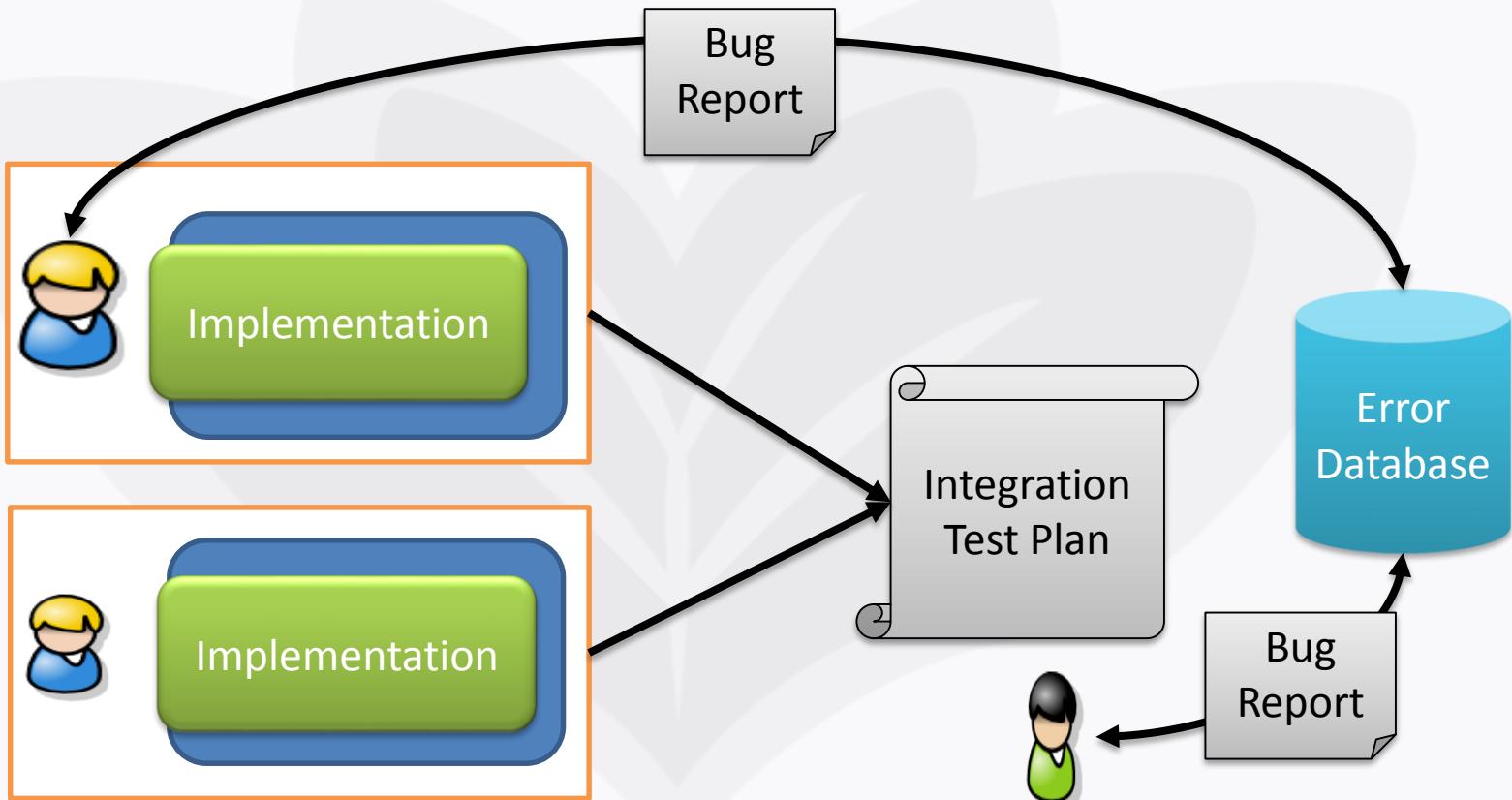


FreeNest

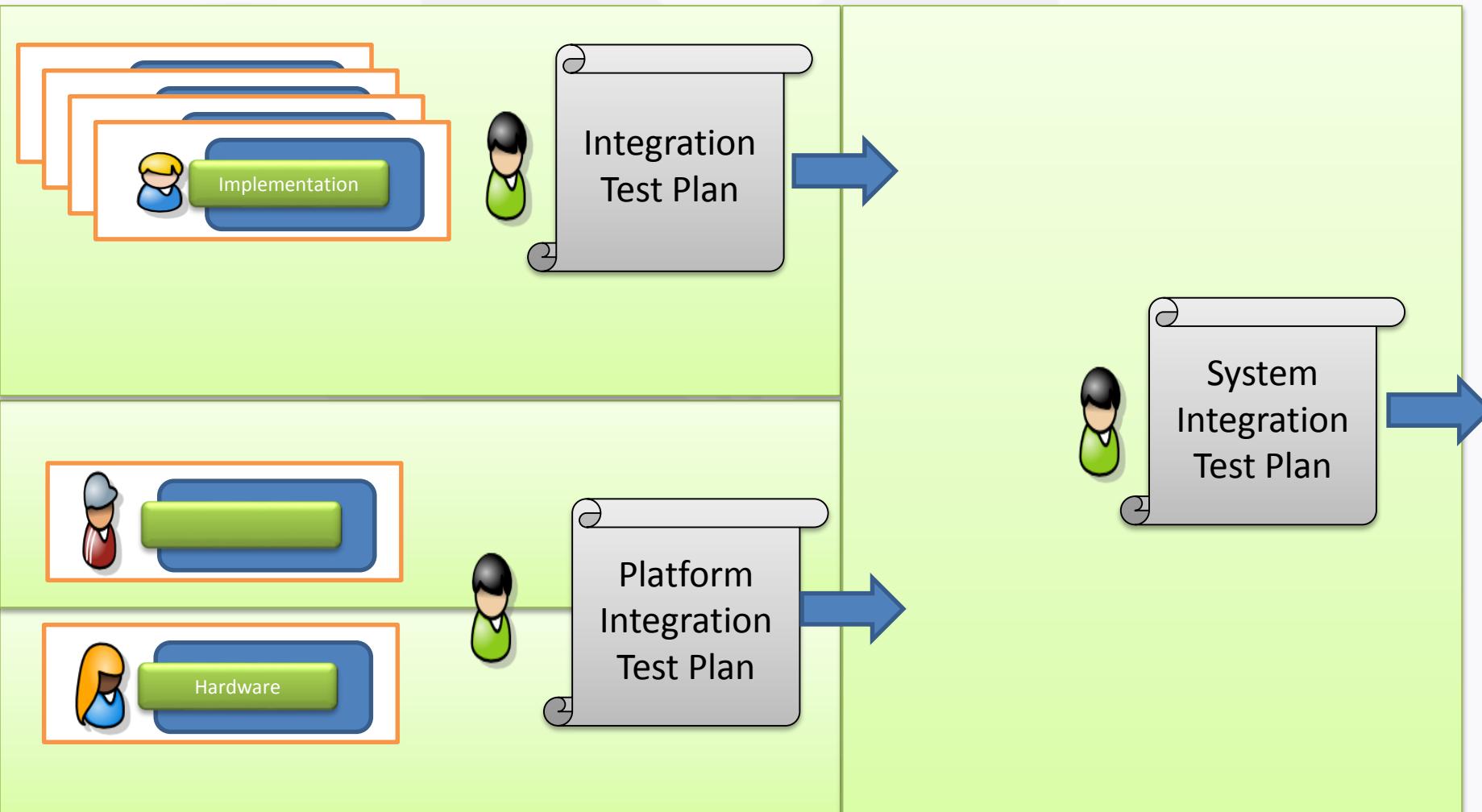
Error Management and unit testing



Error Management and integration testing



Error Management



CMMI Process Framework

CMMI – covers "error management" in several process areas

SP3.2 Analyze Verification results

Typical Work Products: Trouble reports

- Analyze the verification data on defects

- Record all results of the analysis in a report

- Provide information on how defects can be resolved (including verification methods, criteria, and verification environment) and initiate corrective action

Project Monitoring and Control

SG2 Manage Corrective Action to Closure

SP2.1 Analyze Issues

SP2.2 Take corrective Action

SP2.3 Manage corrective Action

SG2 Validate Product or Product Components

SP2.2 Analyze Validation Results. Change request management & Configuration Management process

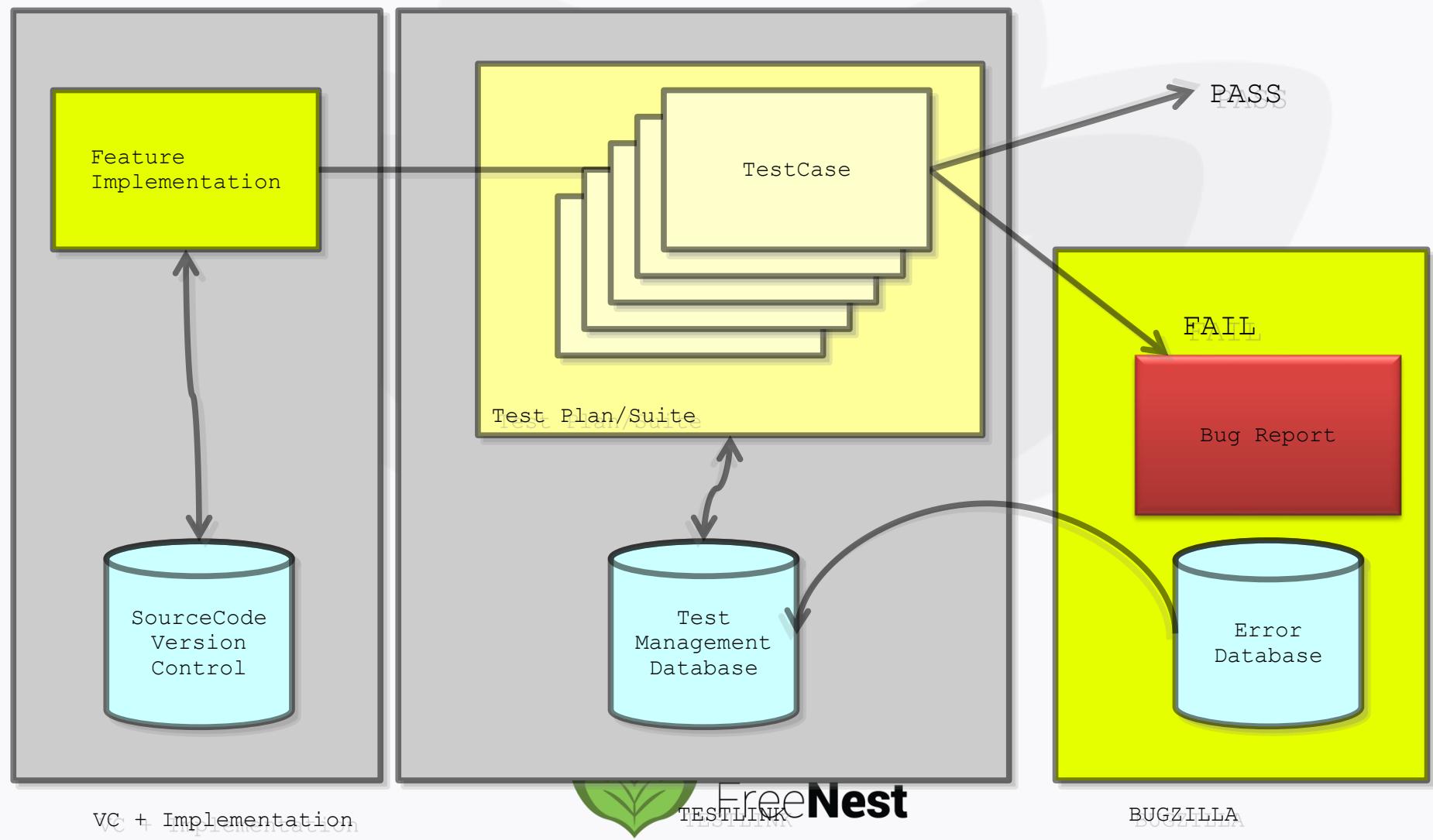
SG2 Track and Control Changes

SP2.1 Track Change Requests

SP2.2 Control Configuration Items

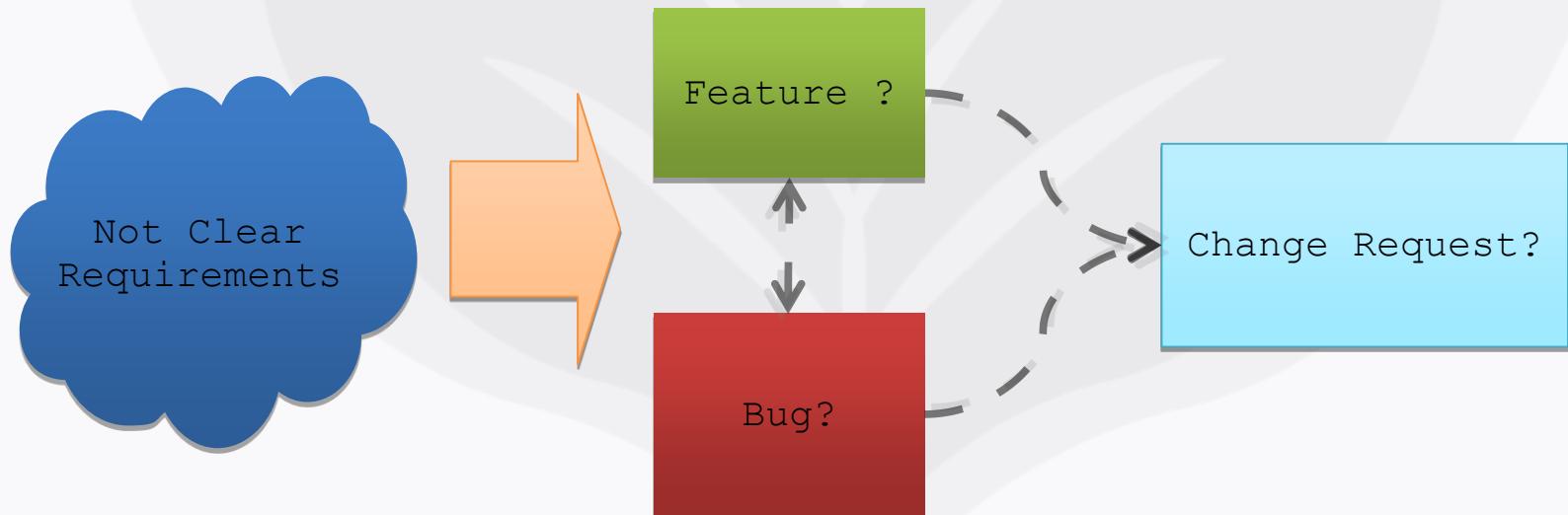


Test & Error Management



Change Management

Sometimes founded defect during testing can lead to change requests



Other issue trackers

JIRA – Commercial

Requisite Pro – Commercial

Rational Synergy - Commercial

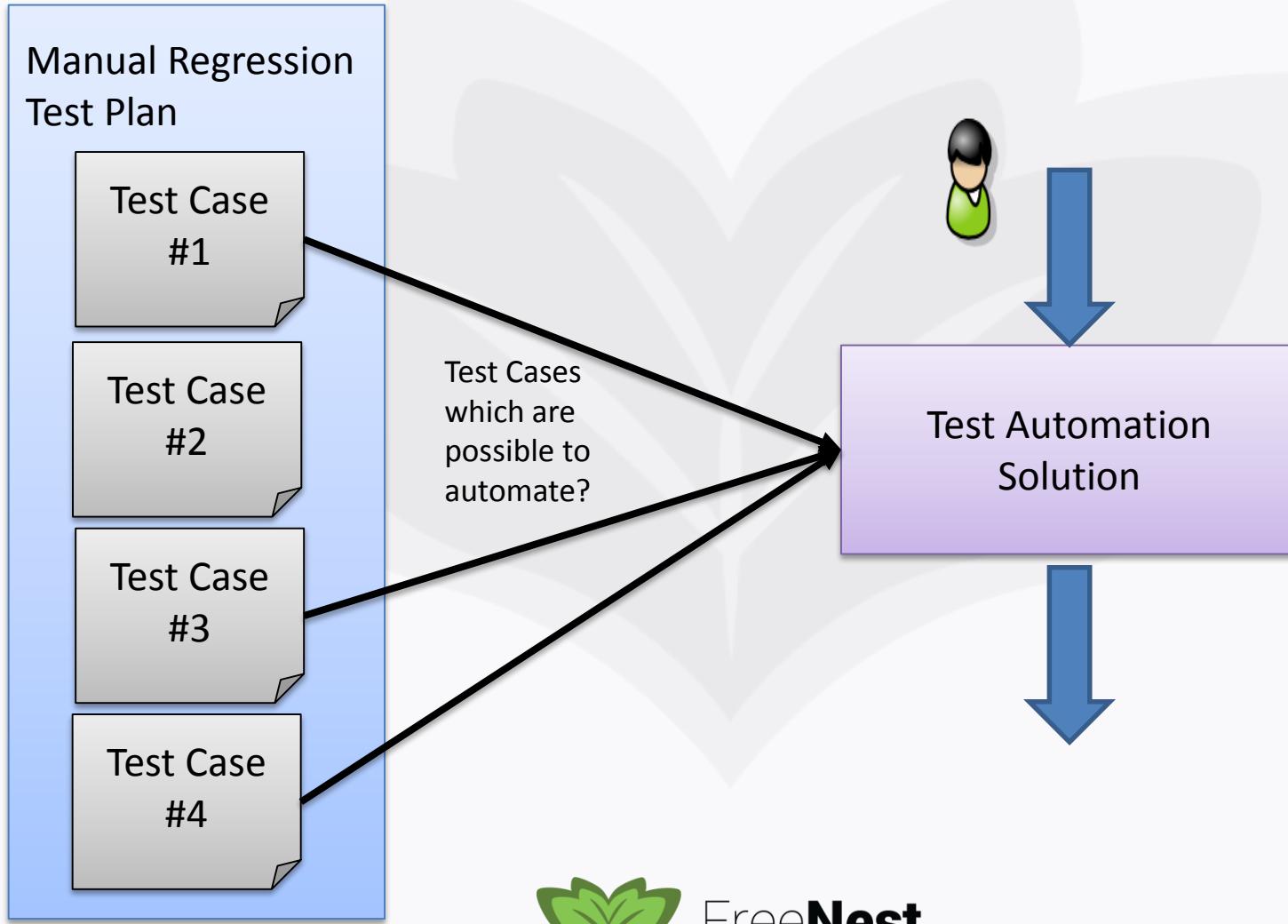
Mantis – Open Source

Regression Testing

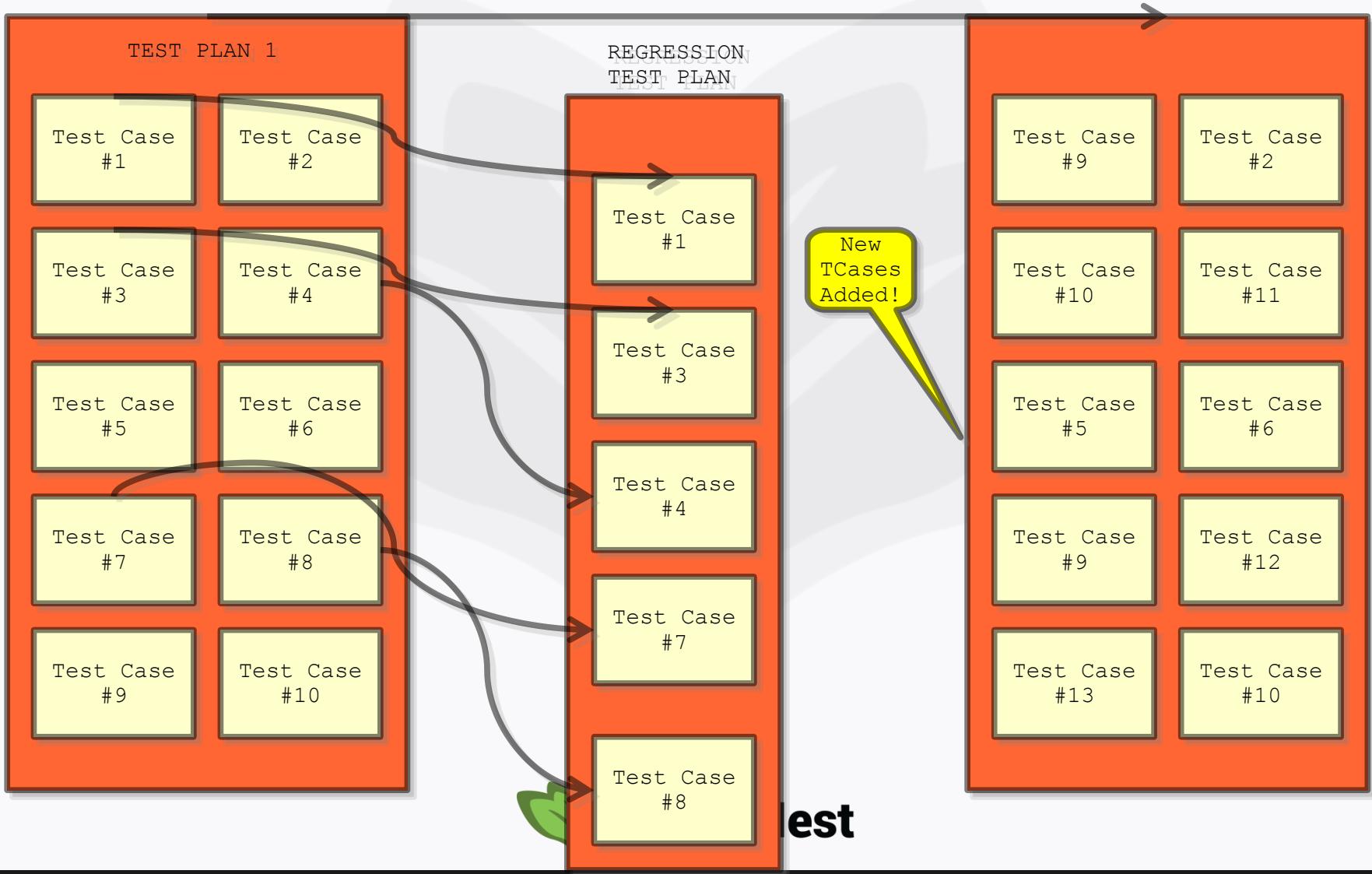
- Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is Changed.

(ISTQB Syllabus)

Regression testing reason for test automation ?



Test Plan & Regression





Team Work 1

- Setup test management solution
- Testlink 1.9.8
- Linux / Windows
- MySQL

Team Work 2

- Setup Issue Management solution
- Bugzilla, Mantis, JIRA, TRAC, Redmine

Team Work 3

- Integration of Test Management & Issue Management
- Testlink + Your Selection

Team Work 4

- Set up CI environment
- Jenkins + SVN/GIT/Mercurial
- Every member of team will add own project in version control tree
- Create Job for all different
- Bonus about static coverage analyze tool installation

Team Work 5

- Install and deploy Robot Framework
- Create test scripts for selected target

Team Work 6

- Do Test Design for selected target
- Design and write 10 Test cases.
- Functionality, Performance, Stability, Release

Team Work 7

- Create Test Planning for selected target
- Do Master Test Plan
- Do Resource Plan
- In Testlink Project

Team Work 8

- Usability testing:
- Read a selected Usability Report
- Analyze current state of solution
- Write report about changes what you can find according report

Team Work 9

- Create performance tester image and plan result gathering and do some performance testing

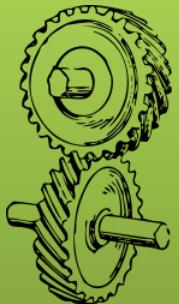
Team Work 10

- Create a video of test case and embed this video as test case in testlink !

Team Work

- Learn how Ubuntu as a product is tested?
- Make presentation about process and show it as a presentation.

Functional Requirement



FUNCTION

Non-Functional Requirement



LOAD

Non-Functional Requirement



SECURITY

Non-Functional Requirement



MAINTENANCE

NON FUNCTIONAL



RECOVERY?

Non-Functional Requirement

```
Main {  
....  
}
```

IMPLEMENTATION

Agile
Epic / Theme

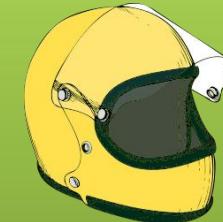


Non-Functional Requirement



SCALING

Non-Functional Requirement



SAFETY

NON FUNCTIONAL



PERFORMANCE?

Team Work



Stake Holders?



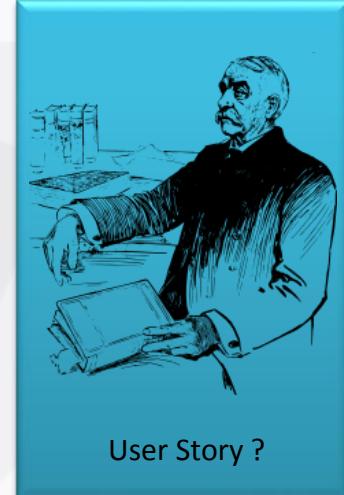
Customer / User ?



Business Request ?



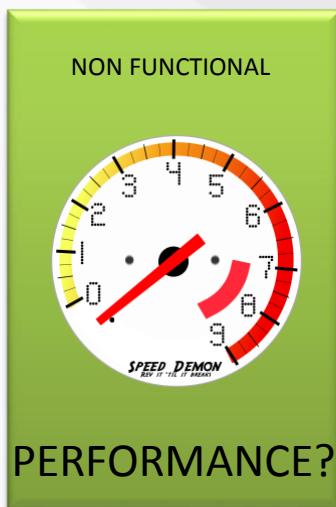
USABILITY



User Story ?



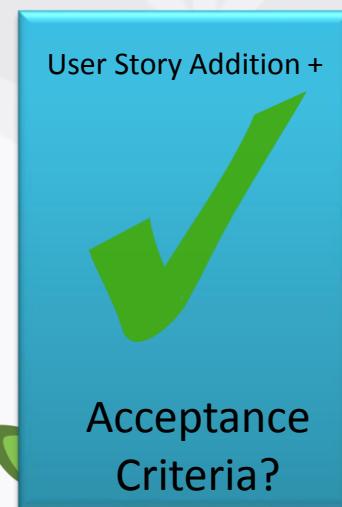
Use Case



NON FUNCTIONAL



PERFORMANCE?



User Story Addition +



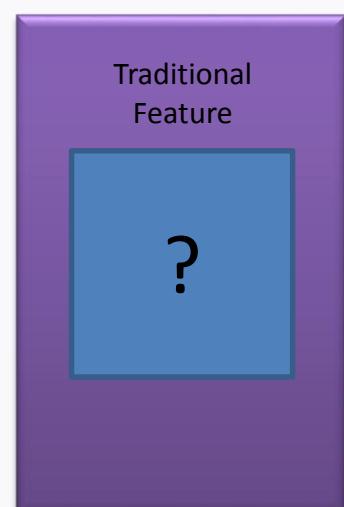
Acceptance
Criteria?



NON FUNCTIONAL



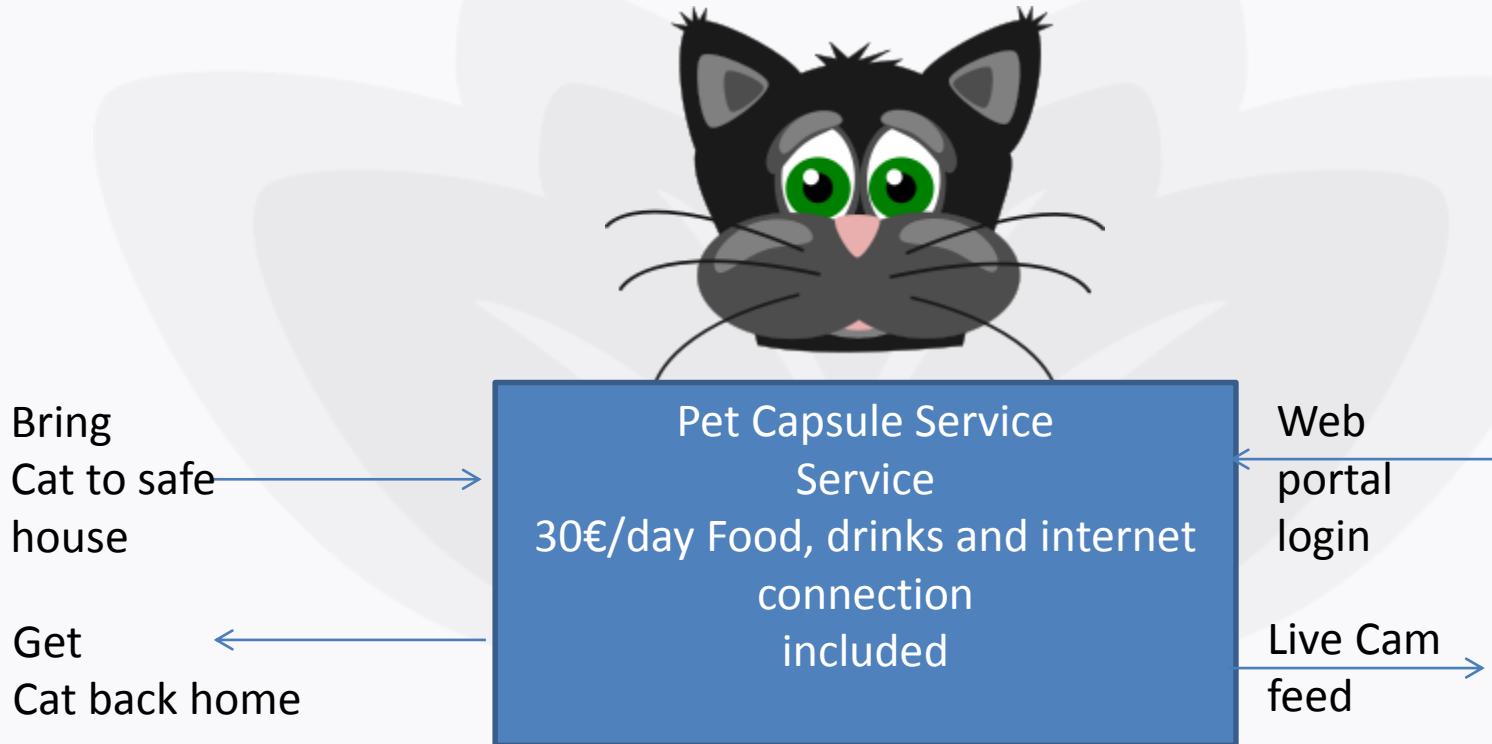
RECOVERY?



Traditional
Feature

?

Pet Capsule Concept



What is The FreeNest Brand?

- We take branding seriously, The FreeNest Brand is our pride and joy.
- It is the message we want to send to the developer teams, customers and stakeholders.
- These guidelines must be followed in all FreeNest's print and electronic communications, marketing and informing.
- Documentation defines logo guidelines, typography and the use of colors.



Logo

- The main logo consists of two elements, the leaf and the text laying horizontally.
- You can also use the logo with the FreeNest text under the leaf, if you need. Don't do this often, the main logo should be the horizontal one.



Free**Nest**



Free**Nest**



Free**Nest**

Logo

- You can also use the logo with dark background and white text.
- You can use the logo with the “Product Platform” text under the FreeNest if the logo itself does not open up in the context.



Logo

Do ...

- observe the free space around the logo.
- use the logo with black text when the background color is lite.
- use the logo with white text when the background color is dark.
- use white, black or neutral backgrounds.
- respect our logos.
- keep the logo readable.

Don't ...

- squash or stretch the logo.
- alter the proportion of the logo.
- alter the layout of the logo.
- put anything over the logo.
- change the colors of the logo.

Brand colors

- TMNT and Leaf Green are the main colors of the brand. You can find the colors from the FreeNest logos leaf. Logos typography has the color called 666. Use these green colors with light or dark background, but do not use TMNT color behind the logo.



TMNT

Hex: #5f8029

CMYK: C67 M30 Y100 K15



Leaf Green

Hex: #8ec03d

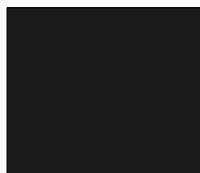
CMYK: C52 M0 Y90 K0



Free**Nest**

Brand colors

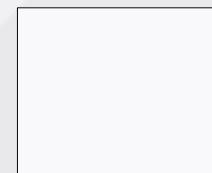
- Kissa can be used as a background color for some elements. It must also be used as a text color if the background is gray (Toolbox for example).
- Gandalf is meant to be used as a text color if the background is light (Steven Seagull).
- Steven Seagull is a main background color for everything in FreeNest. It can be also used as a text color if the background is dark (Kissa for example).



Kissa

Hex: #1a1a1a

CMYK: C76 M67 Y0 K0



Steven Seagull

Hex: #fafafc

CMYK: C2 M2 Y1 K0



Gandalf

Hex: #454545

CMYK: C65 M55 Y52 K53



Free**Nest**

Brand colors

- Sad Cloud, Happy Cloud and Hazy Morning are meant to be used for other visual elements in layouts, buttons, images, illustrations etc. Use them carefully, do not let the blue shades dominate the green shades!



Sad Cloud

Hex: #3b5975

CMYK: C81 M57 Y34 K21



Hazy Morning

Hex: #c5dff7

CMYK: C26 M6 Y0 K0



Happy Cloud

Hex: #7abaf5

CMYK: C53 M16 Y0 K0



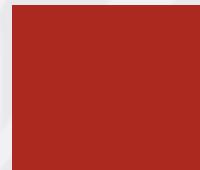
Free**Nest**

Web colors

- Great success, Doing It Wrong and Fatal error are utility colors. They are used to give feedback for the user. Great Success means that everything's good. Doing It Wrong is a notice colour for noticing the user. Fatal Error is used for errors.



Great Success
Hex: #7fb34b



Fatal Error
Hex: #ad2a20



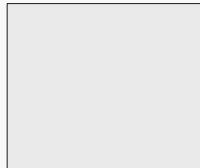
Doing It Wrong
Hex: #ffc30f



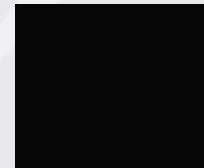
Free**Nest**

Web colors

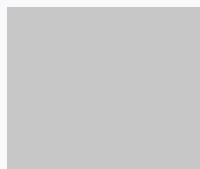
- 666 is the darkest shade you can use anywhere. 666 is near black. If there is necessary to use black, this color is the one.
- Toolbox and Team Board are used as a background for some elements (a toolbox or web site element background for example).



Toolbox
Hex: #ebebeb



666
Hex: #060606



Team Board
Hex: #c8c8c8



Free**Nest**

Typography

The Roboto family

- The brand font is called Roboto. Always maintain the consistency in FreeNest software, web media, print media etc! If it's not possible to use Roboto in web, use Arial.

Roboto Condensed, *Roboto Condensed Italic*

Roboto Bold Condensed, *Roboto Bold Condensed Italic*

Roboto Thin, *Roboto Thin Italic*

Roboto Light, *Roboto Light Italic*

Roboto Regular, *Roboto Regular Italic*

Roboto Bold, *Roboto Bold Italic*

Roboto Black, *Roboto Black Italic*

Typography

Using Roboto

- In body text you should use Thin or Light. Light can be used if Thin gets too thin and stuffy. This text is written with Roboto Light. For highlighting etc. you can use Bold or Black. Never use Italic for anything else than just short quotations and such. See the text colors in the Colors section to select right color for the right background when writing text.

FreeNest (Roboto Thin and Roboto Black)





Free**Nest**

FreeNest Brand



What is The FreeNest Brand?

- We take branding seriously, The FreeNest Brand is our pride and joy.
- It is the message we want to send to the developer teams, customers and stakeholders.
- These guidelines must be followed in all FreeNest's print and electronic communications, marketing and informing.
- Documentation defines logo guidelines, typography and the use of colors.



Logo

- The main logo consists of two elements, the leaf and the text laying horizontally.
- You can also use the logo with the FreeNest text under the leaf, if you need. Don't do this often, the main logo should be the horizontal one.



Free**Nest**



Free**Nest**



Free**Nest**

Logo

- You can also use the logo with dark background and white text.
- You can use the logo with the “Product Platform” text under the FreeNest if the logo itself does not open up in the context.



Logo

Do ...

- observe the free space around the logo.
- use the logo with black text when the background color is lite.
- use the logo with white text when the background color is dark.
- use white, black or neutral backgrounds.
- respect our logos.
- keep the logo readable.

Don't ...

- squash or stretch the logo.
- alter the proportion of the logo.
- alter the layout of the logo.
- put anything over the logo.
- change the colors of the logo.

Brand colors

- TMNT and Leaf Green are the main colors of the brand. You can find the colors from the FreeNest logos leaf. Logos typography has the color called 666. Use these green colors with light or dark background, but do not use TMNT color behind the logo.



TMNT

Hex: #5f8029

CMYK: C67 M30 Y100 K15



Leaf Green

Hex: #8ec03d

CMYK: C52 M0 Y90 K0



Free**Nest**

Brand colors

- Kissa can be used as a background color for some elements. It must also be used as a text color if the background is gray (Toolbox for example).
- Gandalf is meant to be used as a text color if the background is light (Steven Seagull).
- Steven Seagull is a main background color for everything in FreeNest. It can be also used as a text color if the background is dark (Kissa for example).

Kissa

Hex: #1a1a1a

CMYK: C76 M67 Y0 K0

Gandalf

Hex: #454545

CMYK: C65 M55 Y52 K53

Steven Seagull

Hex: #fafafc

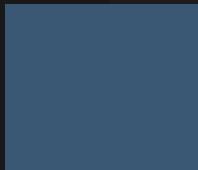
CMYK: C2 M2 Y1 K0



Free**Nest**

Brand colors

- Sad Cloud, Happy Cloud and Hazy Morning are meant to be used for other visual elements in layouts, buttons, images, illustrations etc. Use them carefully, do not let the blue shades dominate the green shades!



Sad Cloud

Hex: #3b5975

CMYK: C81 M57 Y34 K21



Hazy Morning

Hex: #c5dff7

CMYK: C26 M6 Y0 K0



Happy Cloud

Hex: #7abaf5

CMYK: C53 M16 Y0 K0



Free**Nest**

Web colors

- Great success, Doing It Wrong and Fatal error are utility colors. They are used to give feedback for the user. Great Success means that everything's good. Doing It Wrong is a notice colour for noticing the user. Fatal Error is used for errors.



Great Success

Hex: #7fb34b



Fatal Error

Hex: #ad2a20



Doing It Wrong

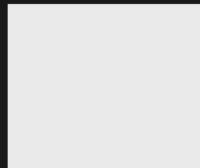
Hex: #ffc30f



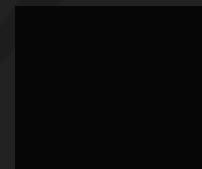
Free**Nest**

Web colors

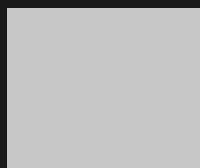
- 666 is the darkest shade you can use anywhere. 666 is near black. If there is necessary to use black, this color is the one.
- Toolbox and Team Board are used as a background for some elements (a toolbox or web site element background for example).



Toolbox
Hex: #ebebeb



666
Hex: #060606



Team Board
Hex: #c8c8c8



Free**Nest**

Typography

The Roboto family

- The brand font is called Roboto. Always maintain the consistency in FreeNest software, web media, print media etc! If it's not possible to use Roboto in web, use Arial.

Roboto Condensed, *Roboto Condensed Italic*

Roboto Bold Condensed, *Roboto Bold Condensed Italic*

Roboto Thin, *Roboto Thin Italic*

Roboto Light, *Roboto Light Italic*

Roboto Regular, *Roboto Regular Italic*

Roboto Bold, *Roboto Bold Italic*

Roboto Black, *Roboto Black Italic*

Typography

Using Roboto

- In body text you should use Thin or Light. Light can be used if Thin gets too thin and stuffy. This text is written with Roboto Light. For highlighting etc. you can use Bold or Black. Never use Italic for anything else than just short quotations and such. See the text colors in the Colors section to select right color for the right background when writing text.

FreeNest (Roboto Thin and Roboto Black)

