EECE 310

Software Maintenance and Evolution

Evolution

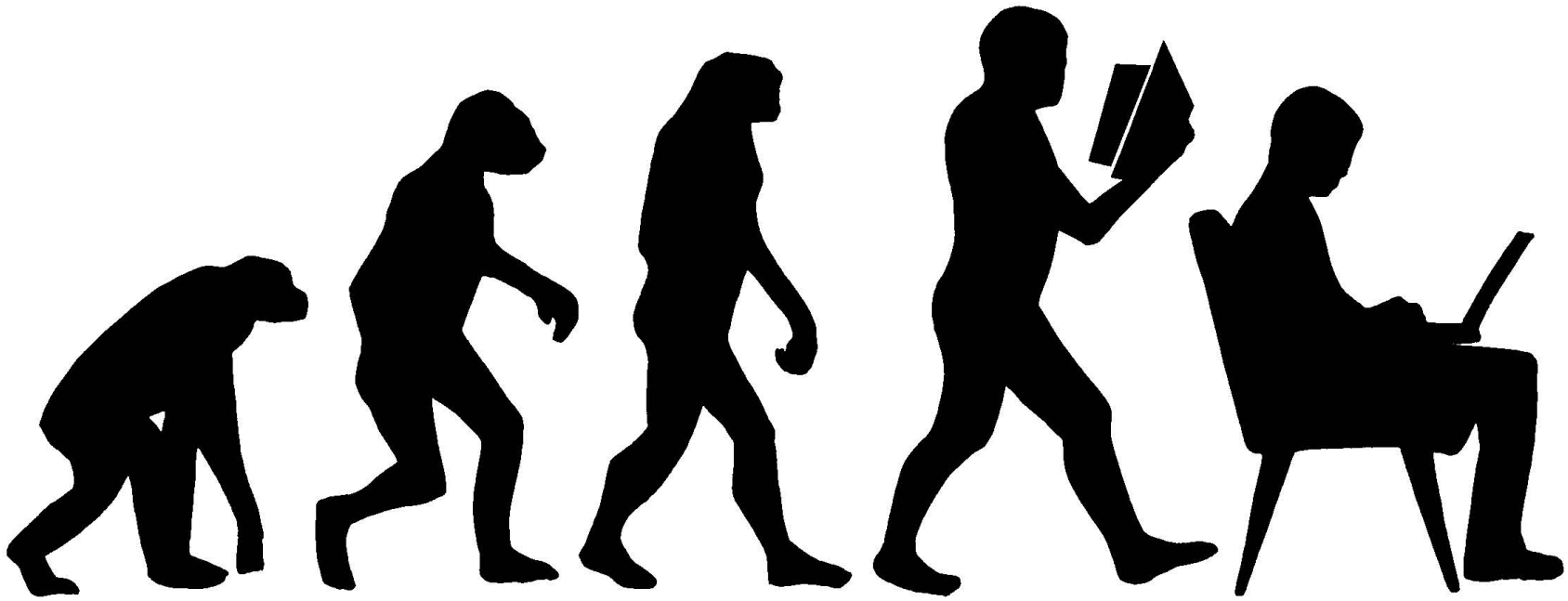


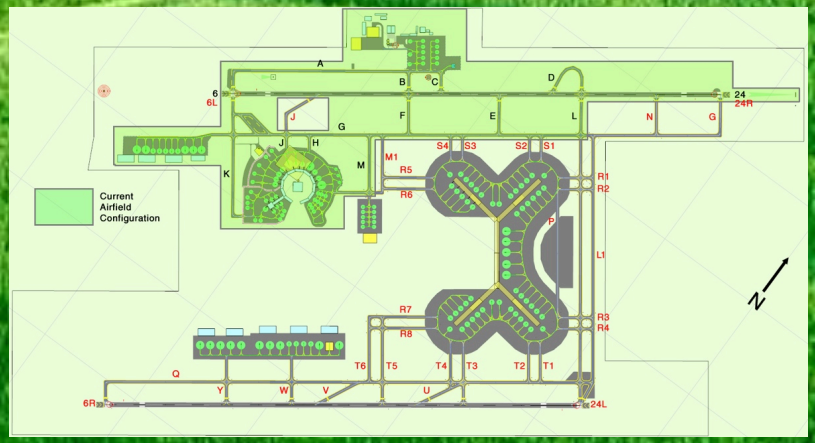
Image from Planet Ivy

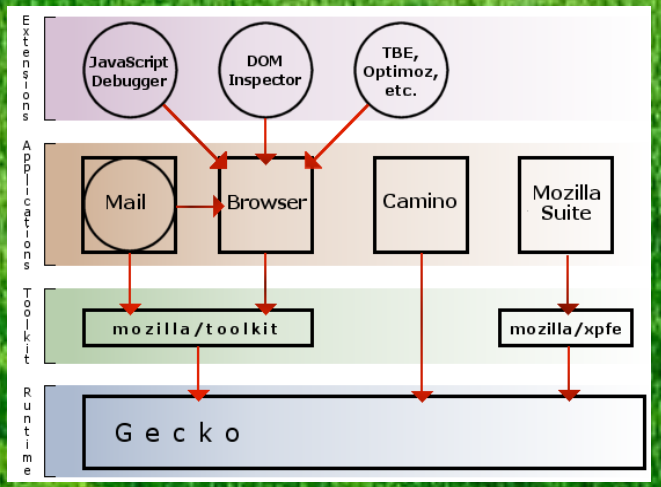
Importance of Evolution?

The gradual development of something, esp. from a simple to a more advanced form.

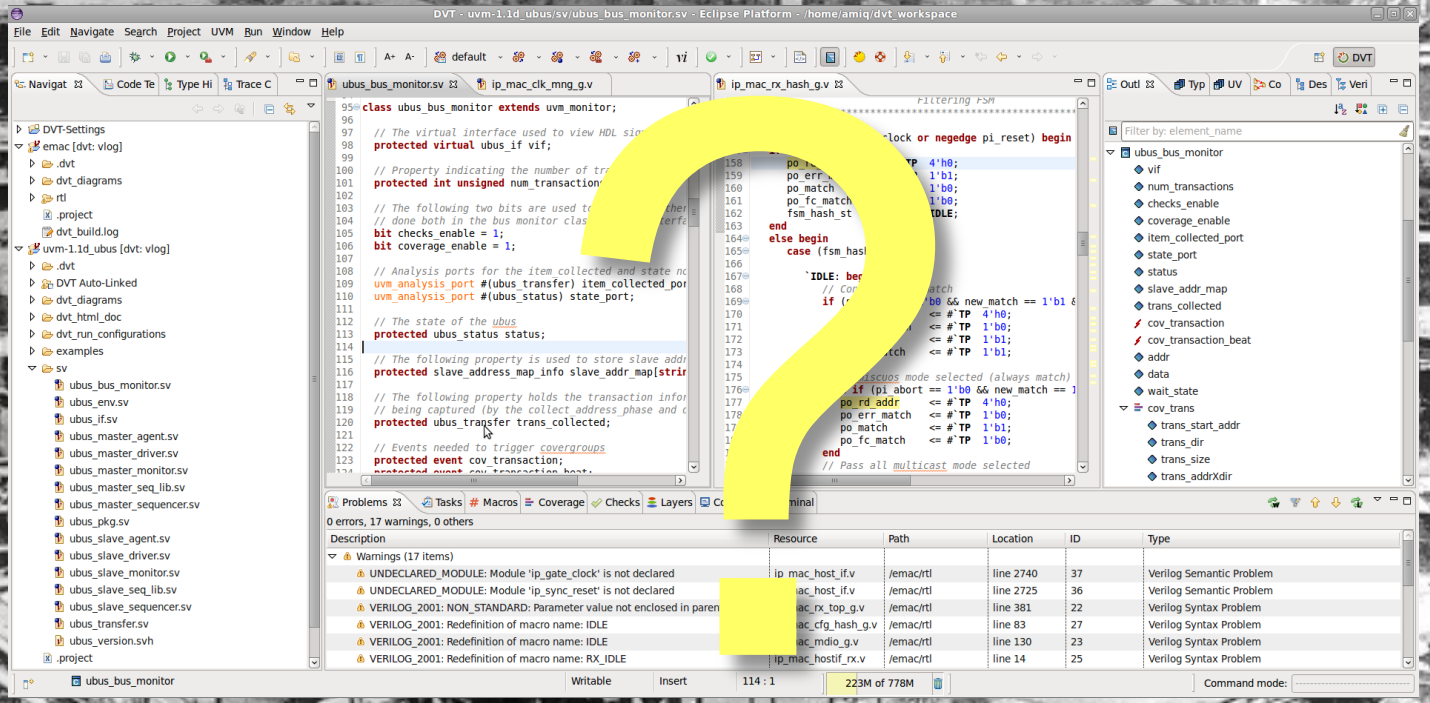
Importance of Maintenance

Greenfield software development





Brownfield software development



Why brownfield engineering?

- Because existing software, often called **legacy software**, is valuable
  - Often business-critical
  - A huge amount of money has already been invested in it
  - Has been tested and runs
  - Does (mainly) what it should do
  - e.g., Banking software (in Cobol)

- Can you replace such a system?

Software Evolution

"All programming activity that is intended to generate a new software version from an earlier operational version"

Manny Lehman and Juan Ramil (2000)

An Evolution Metaphor

Mosaic has structure

So does software

Changing tile by tile

or line by line...

Industrial mosaics?

10 mloc...

> 10 soccer fields

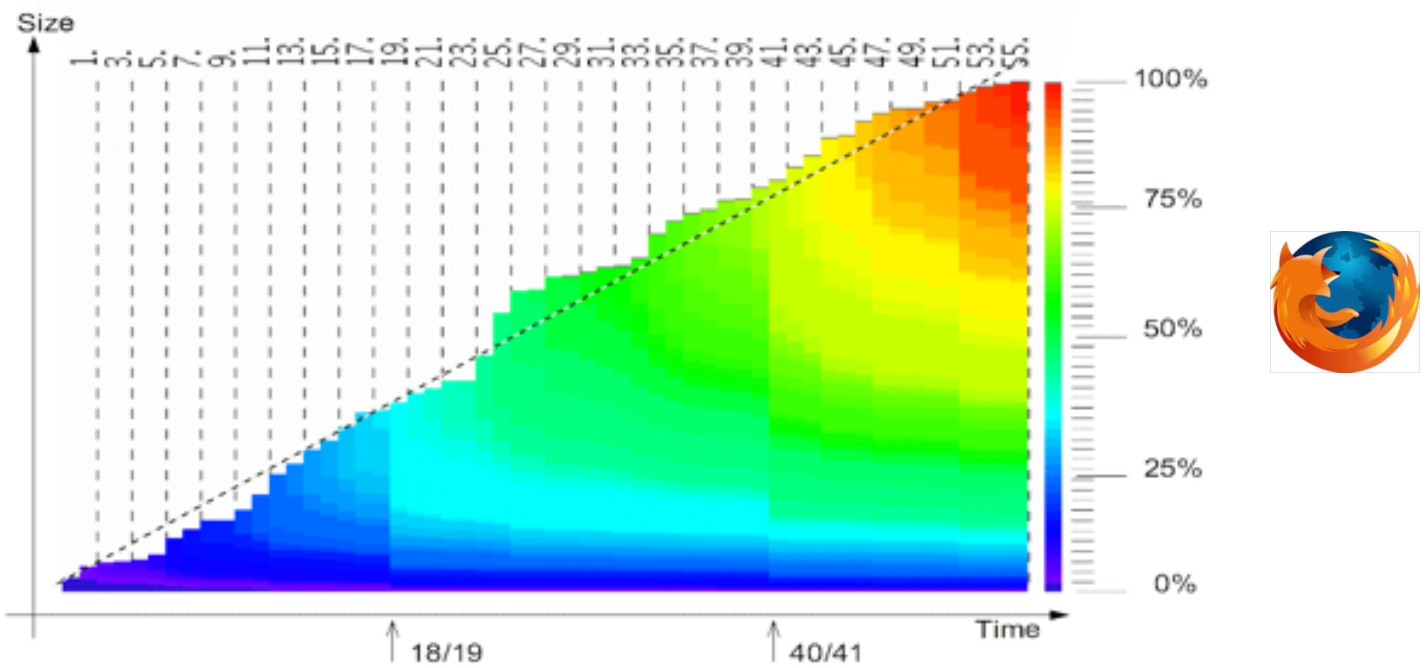
Making changes together:

4 teams of 6 developers ...

moving tiles around.

Evolution of Firefox source code (“Heatmap”)

- 50 releases (0-49) from 1998-04-01 to 2002-06-11
- Nearly linear growth of the project size.
- A small percentage of files have never been modified or updated since they were introduced in the first releases
- More than 50% of the files are modified or added within the last 10 releases (20% of project duration)



Software change

- Software change is inevitable
  - New requirements emerge when the software is used;
  - The business environment changes;
  - Errors must be repaired;
  - New computers and equipment is added to the system;
  - System needs to be maintained;
  - The performance or reliability of the system may have to be improved.

- A key problem for all organizations is implementing and managing change to their existing software systems.

Laws of Software Evolution

- After several major empirical studies at IBM, Lehman proposed that there were a number of 'laws' which applied to all systems as they evolved.

- These are sensible **observations** rather than laws. They are applicable to large systems developed by large organisations.
  - It is not clear if these are applicable to other types of software system.

Laws of Software Evolution

- ***Law of Continuing Change***
  - A program that is used in a real-world environment must necessarily change, or else become progressively less useful in that environment.

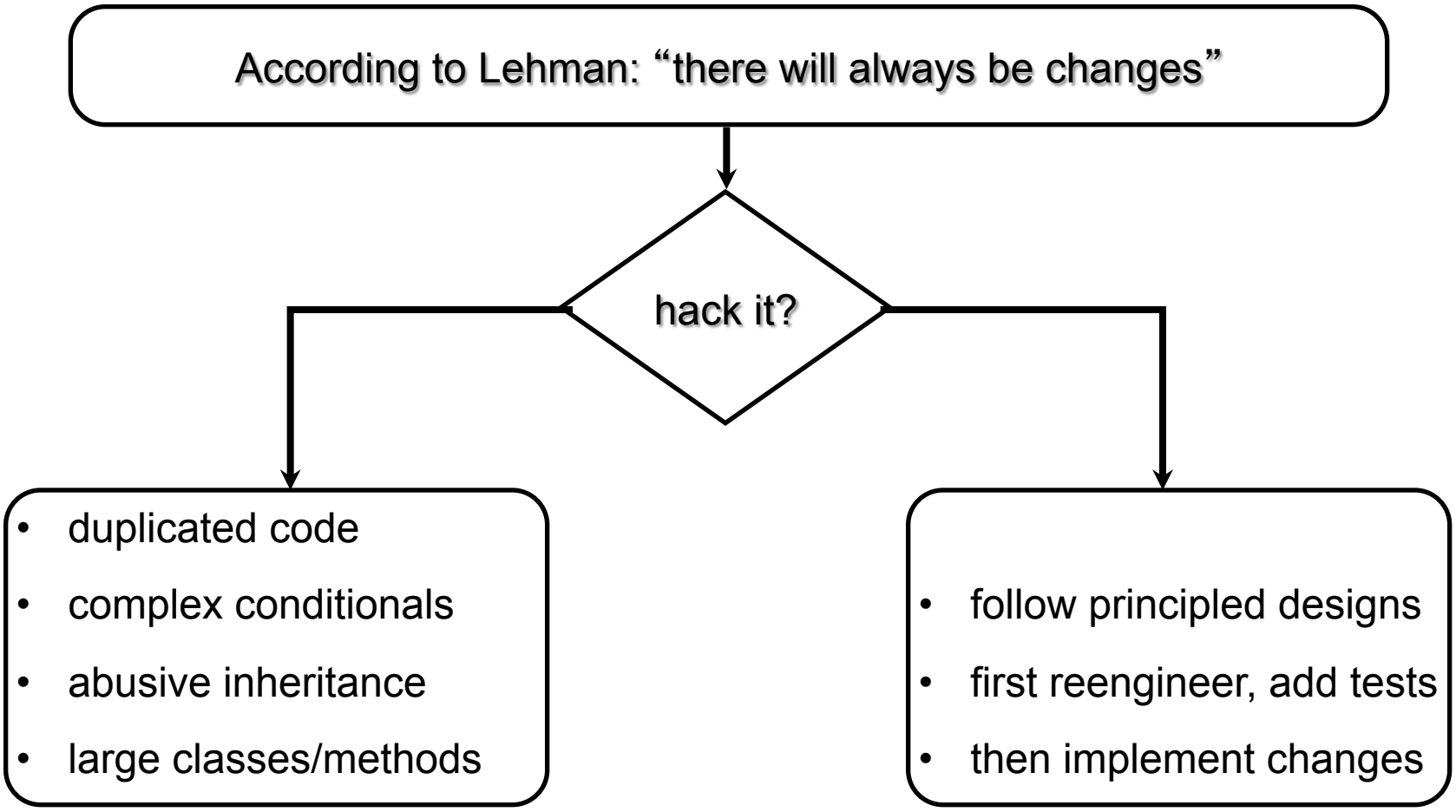
- ***Law of Increasing Entropy***
  - The entropy (aka chaos) of a system increases with time unless specific work is executed to maintain or reduce it.

Lehman's Laws in practice

- Existing software is often modified in an ad-hoc manner (quick fixes)
  - Lack of time, resources, money, etc.
- Initial good design is not maintained
  - Spaghetti code, copy/paste programming, dependencies are introduced, no tests, etc.
- Documentation is not updated (if there is any)
  - Architecture and design documents left to rot
- Original developers leave and with them their knowledge

Result of such practices

What is your decision?



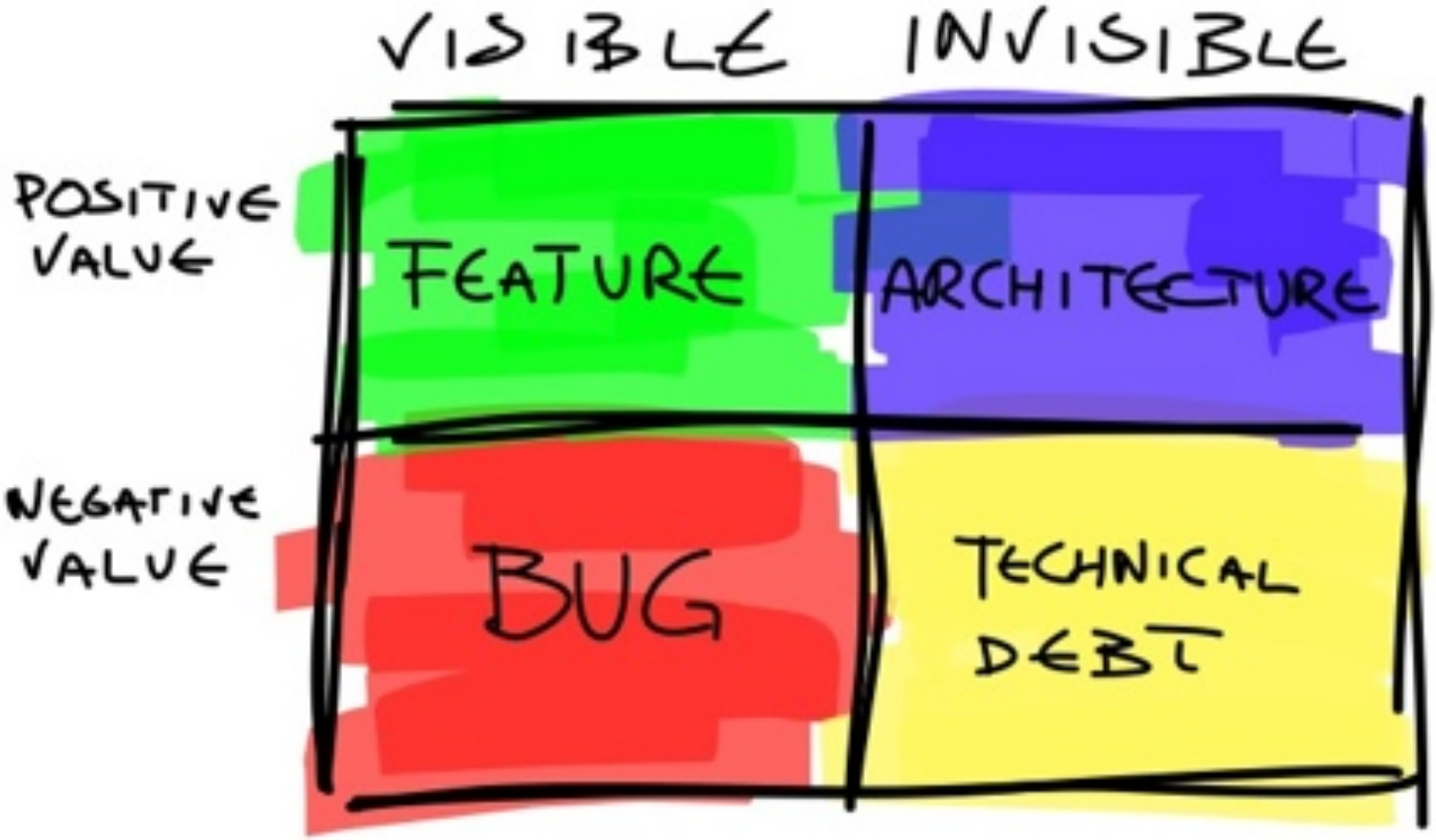
Take a loan on your software, pay back via reengineering (technical debt)

Investment for the future, paid back during maintenance

Technical Debt

"Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite. The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt."

-Ward Cunningham, the programmer who developed the first wiki program



Technical debt sounds a lot like security weaknesses. Invisible from users but has negative value.

Technical Debt

Issues found in the code that will affect future development but not those dealing with feature completeness.

Examples

- Code smells: 167 person hours
- Missing tests: 298 person hours
- Bad Design: 670 person hours
- Documentation: 67 person hours

Totals

- Work: 1,202 person hours (50 person days)
- Cost: $57,000

Debt is Good!

- There are obviously good business reasons for accumulating security debt because we see it everywhere in successful companies.
- **However**, there is a point in the lifetime of software projects where the debt gets too *high* and needs to be paid off by **redesigning** and **rewriting** a lot of code.
- If it isn't paid off the debt risks impacting the bottom line.
- e.g., Vista - > Windows 7

HERE

Laws of Software Evolution

- ***Law of Continuing Change***
  - A program that is used in a real-world environment must necessarily change, or else become progressively less useful in that environment.

- ***Law of Increasing Entropy***
  - The entropy (aka chaos) of a system increases with time unless specific work is executed to maintain or reduce it.

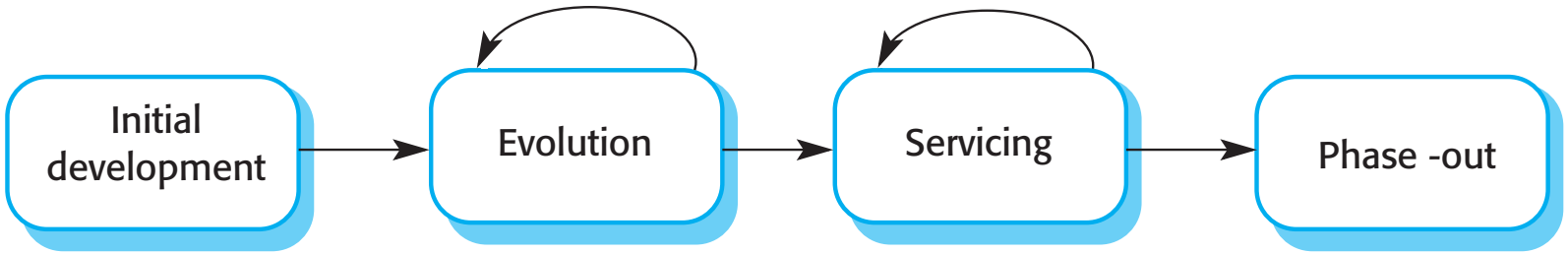
In-class exercise

Form your lab groups,

1. Discuss the rationale behind the laws of software evolution.

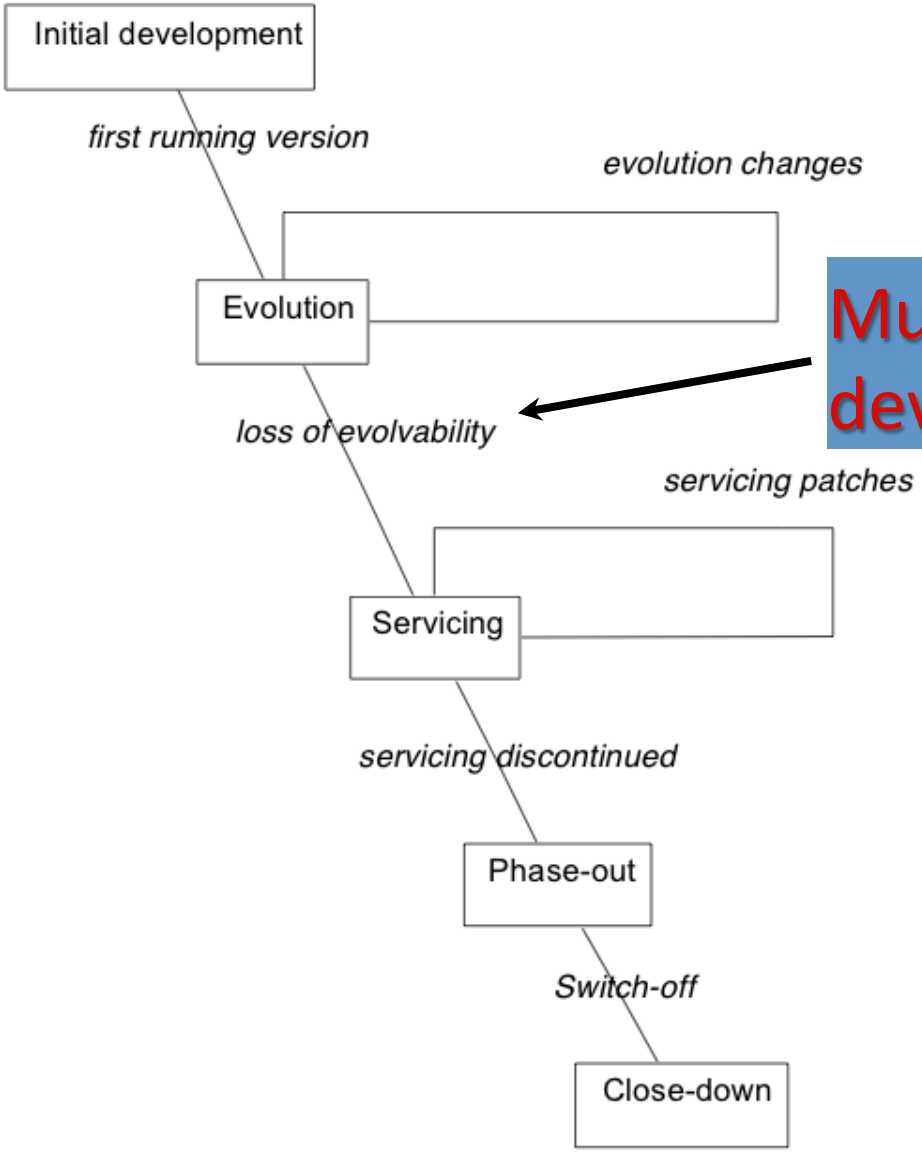
2. Under what circumstances might the laws break down?

Different stages of software



Stages

- Evolution
  - The stage in a software system's life cycle where it is in operational use and is evolving as **new requirements** are proposed and implemented in the system.
- Servicing
  - At this stage, the software remains useful but the only changes made are those **required** to keep it operational i.e. bug fixes and changes to reflect changes in the software's environment. No new functionality is added.
- Phase-out
  - The software may still be used but **no further changes** are made to it.



Must retain product and developer knowledge

Figure 1. The simple staged model

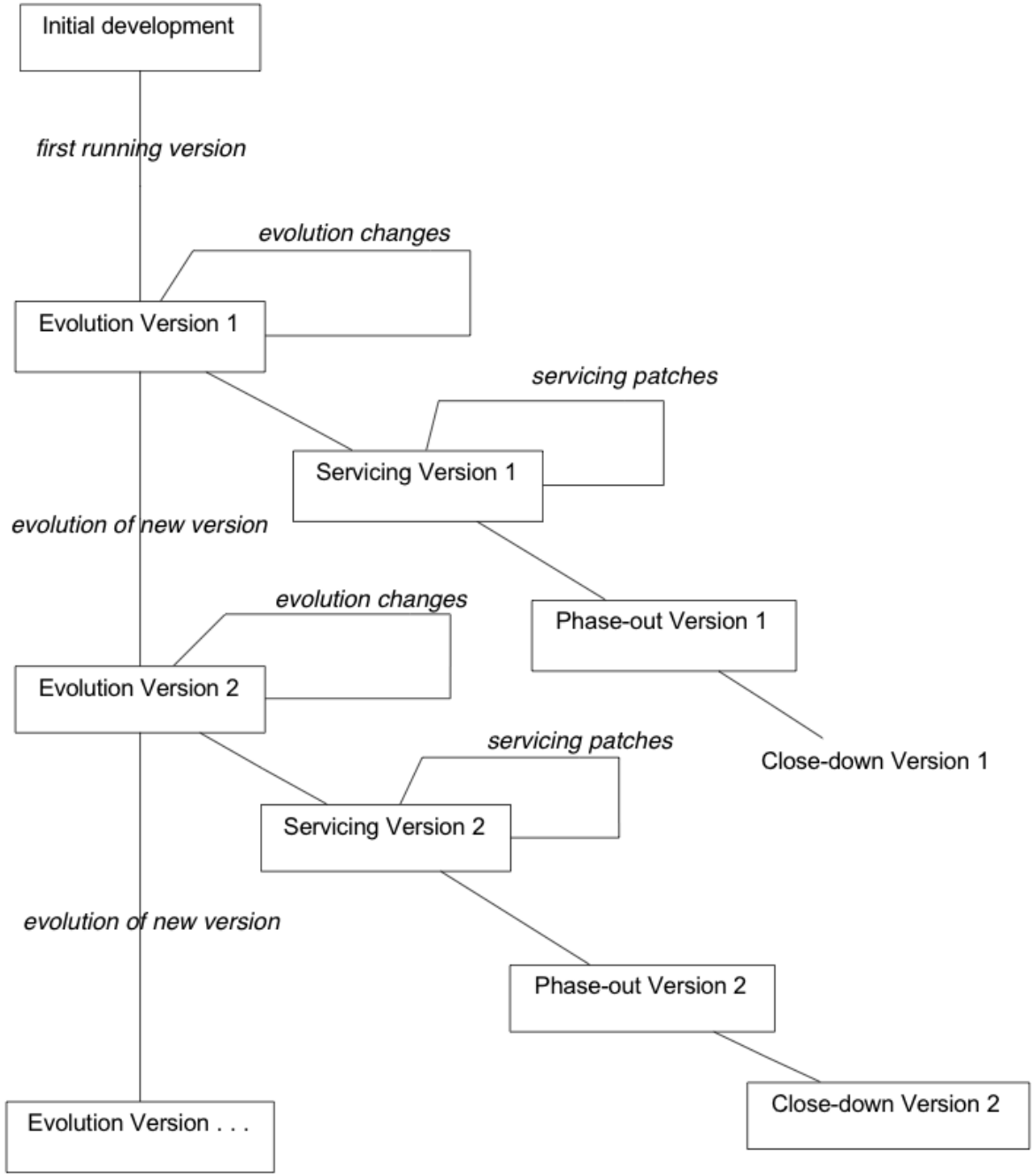


Figure 2. The versioned staged model

Evolution processes

- Proposals for change are the driver for system evolution.
  - Should be linked with components that are affected by the change, thus allowing the cost and impact of the change to be **estimated**.
  - **Change impact analysis**

- Change identification and evolution continues throughout the system lifetime.

What is Software Maintenance?

- Producing new (versions of) software under the constraints of deployed software
  - Backwards compatibility is often required
  - aka "Brownfield development"
  - Legacy Software: "software which is vital to our organization, but we don't know what to do with it"

Similar to development, but HARDER!

Maintenance: where do we start?

- Software maintenance can comprise all phases of the lifecycle, starting with **requirements gathering**

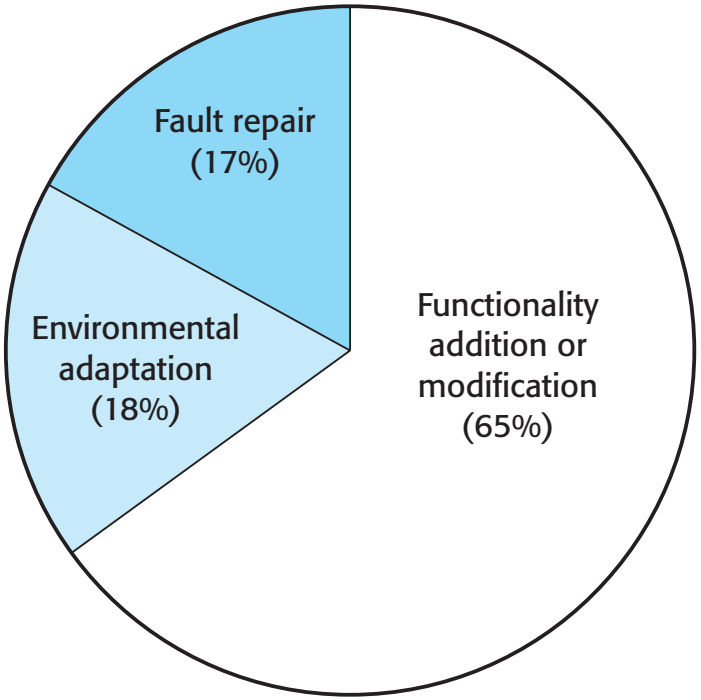
Agile methods and evolution

- Agile methods are based on incremental development so the transition from development to evolution is a seamless one.
  - Evolution is simply a continuation of the development process based on frequent system releases.
- Automated **regression testing** is particularly valuable when changes are made to a system.
- Changes may be expressed as additional user stories.

Y2K Problem

- Memory space used to be a problem (still is in a lot of places).
- So, store 2 digit years
  - (will your code be still in use in the year 2100?).
- Approaching the year 2000 (millennium problem)
  - Systems would crash, calculations could go wrong, PANIC!
- Mitigation: efforts cost ~ $300billion worldwide.
- A lot of SE jobs were created and some got rich(er)!

Maintenance effort distribution?



Types of maintenance

- **Corrective**
  - correct faults in system behaviour
  - caused by errors in code, design, SRS
- **Adaptive**
  - due to changes in operating environment
- **Perfective**
  - due to changes in requirements
  - “pivots” for business needs (new features, etc)
- **Preventive**
  - dealing with legacy problems, refactoring, clean code

Question

Implement an alarm clock function

Opened 1 week ago
Last modified 2 days ago

Reported by:	sebokie@hotmail.fr	Assigned to:	varun
Priority:	3	Milestone:	1.5 "Lothlórien"
Component:	(Jajuk Members) Any (Default Component)	Version:	1.4
Keywords:		Cc:	

Description — Reply

Hello and thanks for that great software.

Would it be possible to implement an alarm clock function in Jajuk, as it would be one of the only players to do this and I'm convinced it would be used by many other persons than me!

Thanks and keep up with the good work!

A. Perfective C. Adaptive

B. Corrective D. Preventive

Question

Implement an alarm clock function

Opened 1 week ago
Last modified 2 days ago

Reported by:	sebokie@hotmail.fr	Assigned to:	varun
Priority:	3	Milestone:	1.5 "Lothlórien"
Component:	(Jajuk Members) Any (Default Component)	Version:	1.4
Keywords:		Cc:	

Description — Reply

Hello and thanks for that great software.

Would it be possible to implement an alarm clock function in Jajuk, as it would be one of the only players to do this and I'm convinced it would be used by many other persons than me!

Thanks and keep up with the good work!

A.

What type is this?

same Jajuk database/config/cache working accross different systems and different platforms Opened 4 months ago

Reported by:	fsck222	Assigned to:	bflorat
Priority:	5	Milestone:	1.5 "Lothlórien"
Component:	(Java Developer) Functional	Version:	1.3.10
Keywords:		Cc:	

Description Reply

from discussion on ~~#318~~

I would like to have the same Jajuk database, some of the config, the cache, etc... working across different systems and different platforms. The user case is the following: You have an external hard drive, a desktop PC under Windows at work, a desktop PC under Linux at home. You want to have your music and your Jajuk files (database, cache, config, perspective, etc...) only on your external hard drive. I believe the best way is to use alternate path in the devices configuration. I think we should start to use profiles to separate hardware configurations options from the others. My propositions is described below:

- In the database, in the device declaration section, we need to have a list of alternate path of the device. Those alternate path could be configured in the device property box.
- When mounting a device, Jajuk will try using the first path of the list and if he can't it will try the second one, and etc...

> Not sure it's faisable as device id is computed on the raw name but, please create a feature, we'll check that later. For windows drive letter issue (letter can change according to connected devices, we have a solution: we can map a letter to a device using some advanced parameters/ storage options)

What type is this?

same Jajuk database/config/cache working accross different systems and different platforms Opened 4 months ago

Reported by:	fsck222	Assigned to:	bflorat
Priority:	5	Milestone:	1.5 "Lothlórien"
Component:	(Java Developer) Functional	Version:	1.3.10
Keywords:		Cc:	

Description Reply

from discussion on ~~#318~~

I would like to have the same Jajuk database, some of the config, the cache, etc... working across different systems and different platforms. The user case is the following: You have an external hard drive, a desktop PC under Windows at work, a desktop PC under Linux at home. You want to have your music and your Jajuk files (database, cache, config, perspective, etc...) only on your external hard drive. I believe the best way is to use alternate path in the devices configuration. I think we should start to use profiles to separate hardware configurations options from the others. My propositions is described below:

- In the database, in the device declaration section, we need to have a list of alternate path of the device. Those alternate path could be configured in the device property box.
- When mounting a device, Jajuk will try using the first path of the list and if he can't it will try the second one, and etc...

> Not sure it's faisable as device id is computed on the raw name but, please create a feature, we'll check that later. For windows drive letter issue (letter can change according to connected devices, we have a solution: we can map a letter to a device using some advanced parameters/ storage options)

Adaptive

And this?

track progression not displayed properly for VBR mp3s

Opened 2 months ago
Last modified 1 month ago

Reported by:	dxnihilo@yahoo.com	Assigned to:	bflorat (accepted)
Priority:	5	Milestone:	1.5 "Lothlórien"
Component:	(Jajuk Members) Any (Default Component)	Version:	1.3.11
Keywords:		Cc:	

Description

Reply

The track progression bar does not display properly for variable bitrate mp3 files. It usually adds anywhere from about 20 seconds to 5 minutes to the real length of the track, although in one case it actually subtracted some time from the length (and the track stopped playing at that shorter length). Also, the track position slider does not work properly for these files. If I let a track play through without moving the slider, it ends at the proper time, but if I move the slider it does not skip to the corresponding position in the track (part of the track can play more than once). The time remaining field displays properly but if the slider is moved past the real track length it displays 0:00 while part of the track is still playing.

And this?

track progression not displayed properly for VBR mp3s

Opened 2 months ago
Last modified 1 month ago

Reported by:	dxnihilo@yahoo.com	Assigned to:	bflorat (accepted)
Priority:	5	Milestone:	1.5 "Lothlórien"
Component:	(Jajuk Members) Any (Default Component)	Version:	1.3.11
Keywords:		Cc:	

Description — Reply

The track progression bar does not display properly for variable bitrate mp3 files. It usually adds anywhere from about 20 seconds to 5 minutes to the real length of the track, although in one case it actually subtracted some time from the length (and the track stopped playing at that shorter length). Also, the track position slider does not work properly for these files. If I let a track play through without moving the slider, it ends at the proper time, but if I move the slider it does not skip to the corresponding position in the track (part of the track can play more than once). The time remaining field displays properly but if the slider is moved past the real track length it displays 0:00 while part of the track is still playing.

Corrective

Maintenance strategies

- Corrective
  - typically a maintenance contract or policy
  - error-reporting and audits
- Perfective
  - Adopt an iterative development strategy (agile?)
- Adaptive/Preventive
  - Try to anticipate, monitor and manage

In Reality, Maintenance...

- … is what you will do most of the time.

- … is often done as an **afterthought**, turning it into a nightmare.
  - You have to think ahead of time how you (or someone else) are going to maintain the code later.
  - Refactoring is crucial
    - ... but it should be done regularly and must start early

Maintenance stats

- Maintenance consumes **40-80**% of software budget
- Typical developer activity (Swanson 1980)
  - 48% maintenance, 46% new development
  - A lot of legacy code to maintain (e.g. Cobol used in banks)

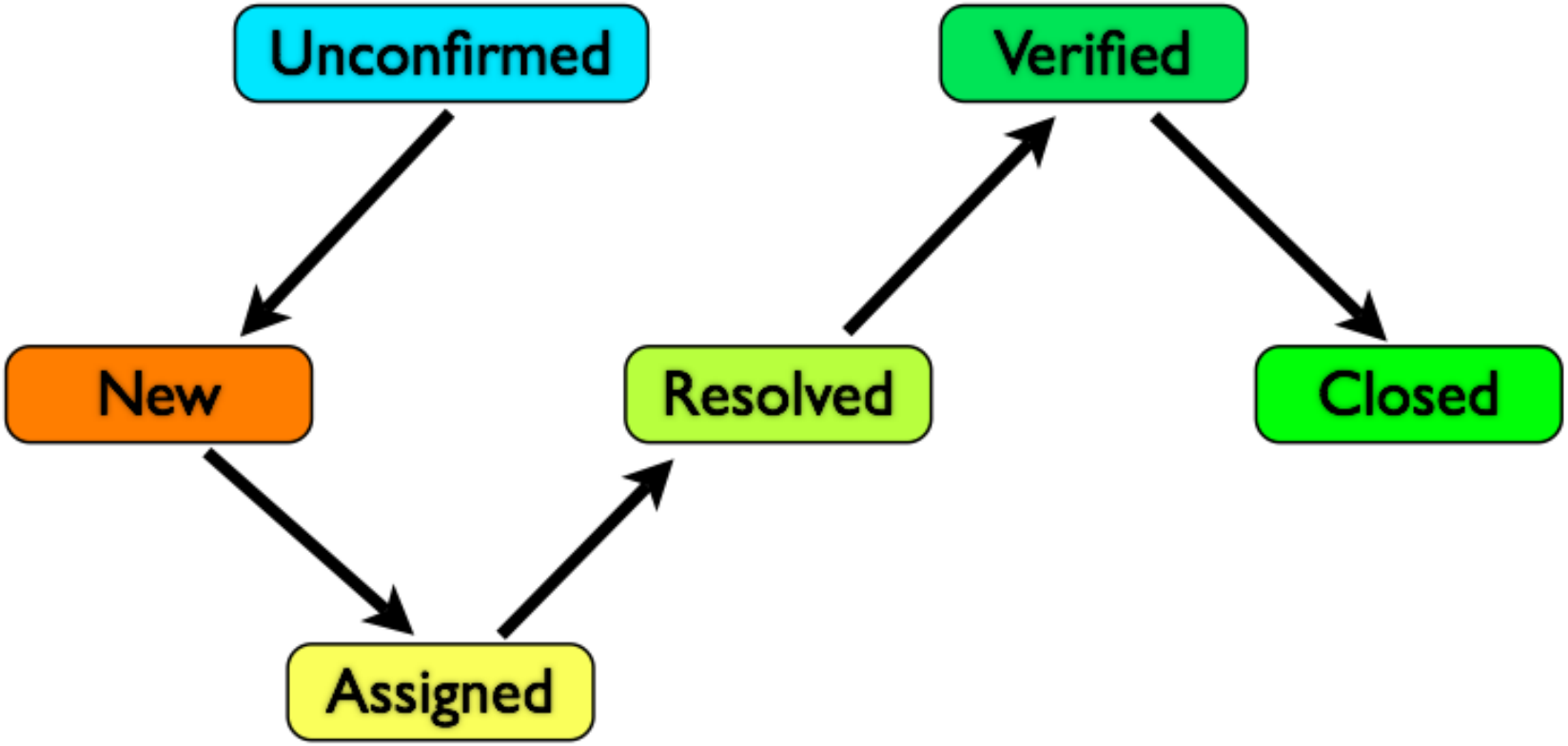
Why is maintenance hard?

- **Lack of knowledge of the application domain**
  - —understanding the implications of change
- **Lack of test suites**
- **Poor code quality to understand**
  - —opaque code
  - —poorly structured code
  - —dead code
- **Lack of documentation**
  - —code is often the only resource
  - —missing rationale for design decisions

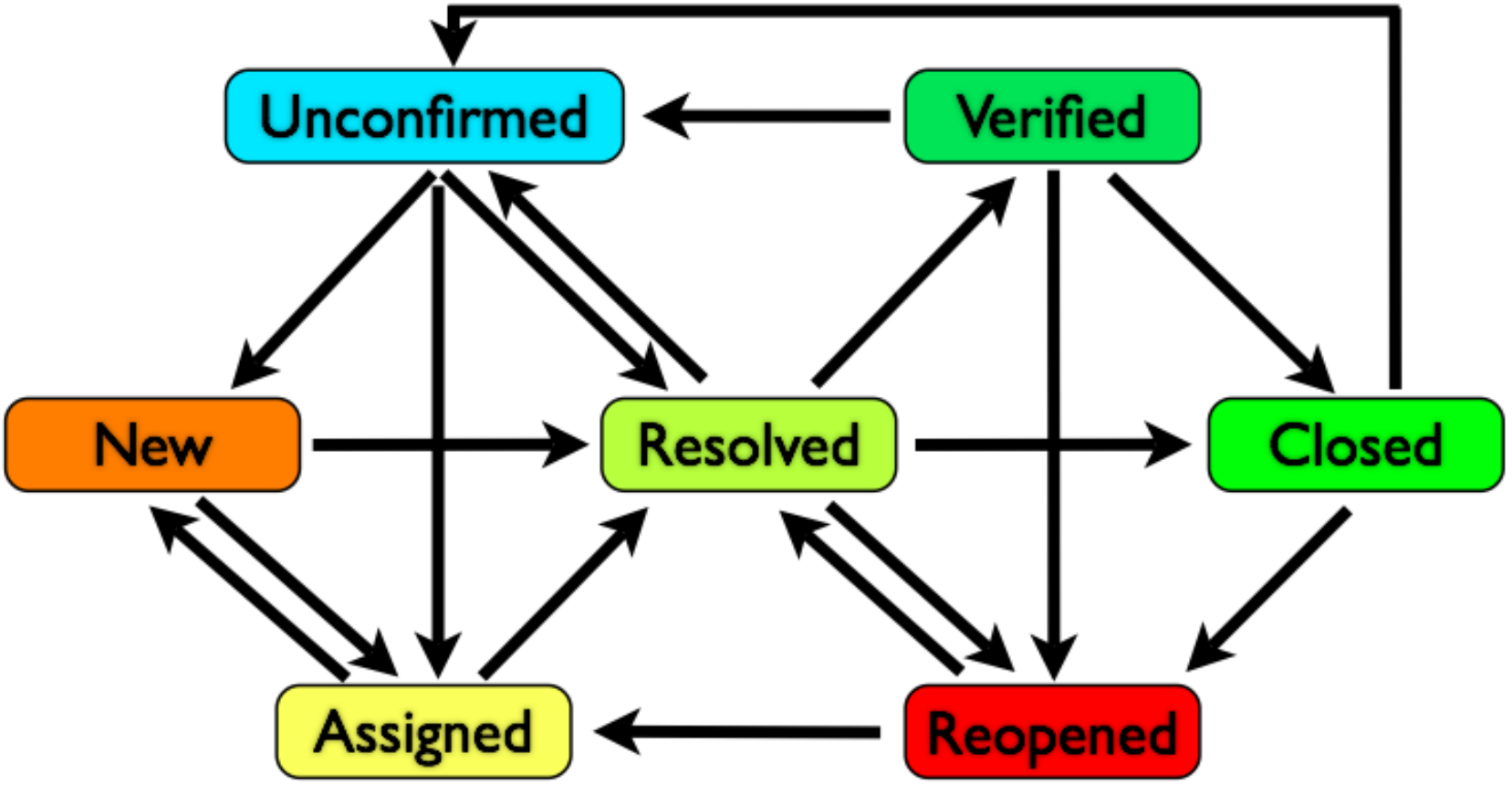
What is the first thing you do when you find a software bug?

- Make sure it is really a bug and **file a bug report**!

An ideal bug's life cycle



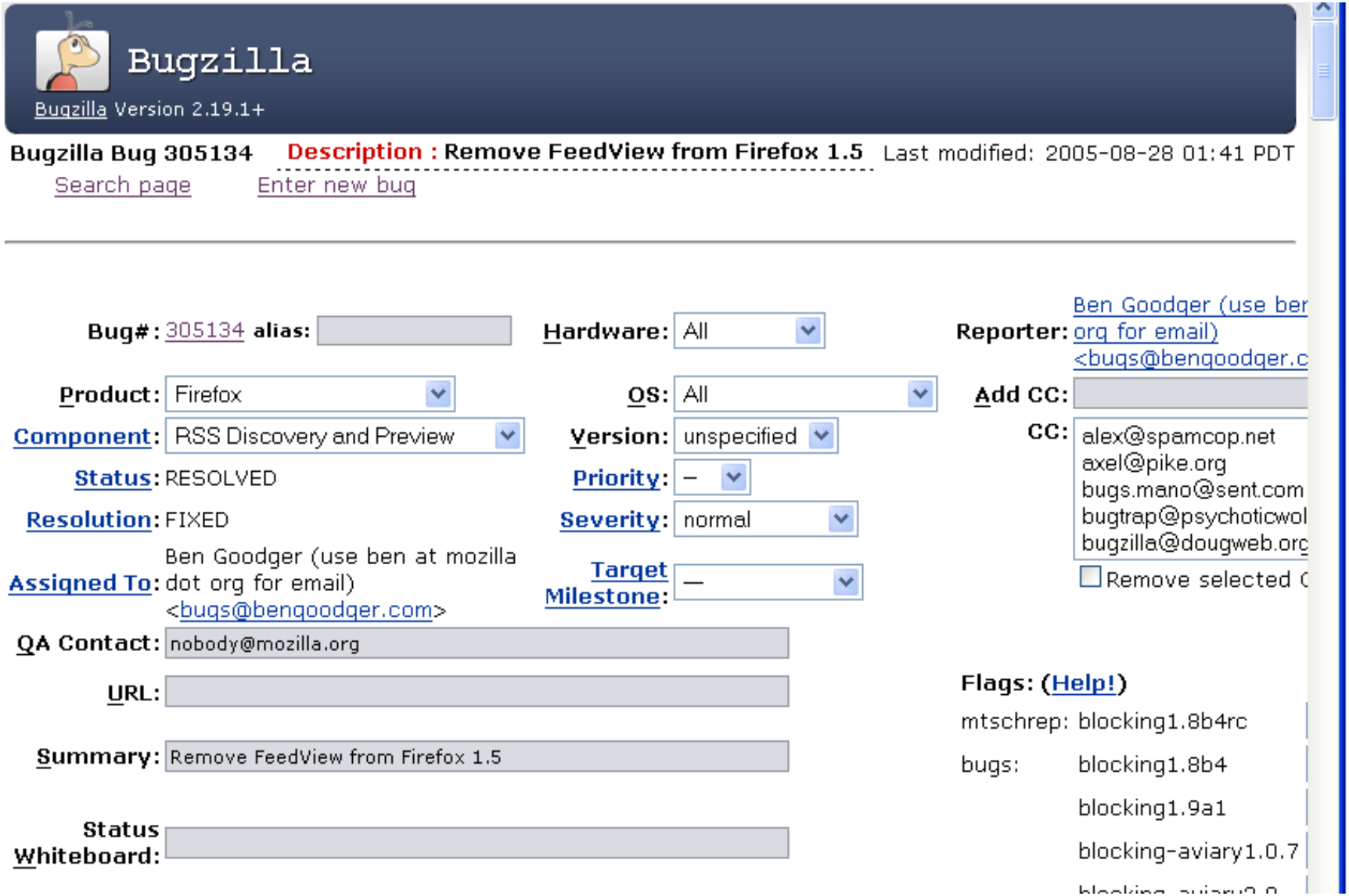
A real bug's life cycle



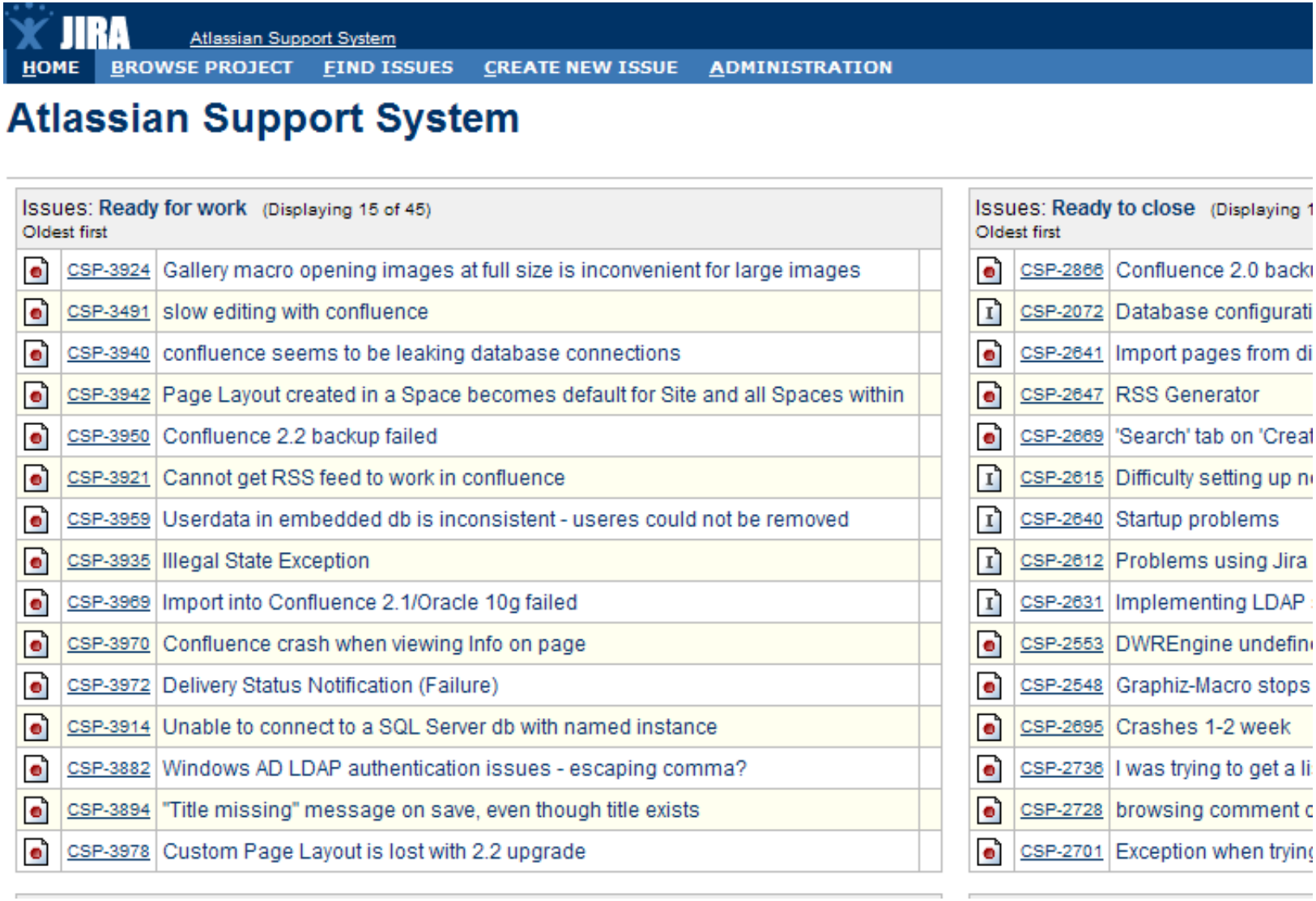
Bug Tracking

- GitHub Issues
- Bugzilla
- Jira
- Trac

One bug


Bugzilla
Bugzilla Version 2.19.1+
Bugzilla Bug 305134 Description : Remove FeedView from Firefox 1.5 Last modified: 2005-08-28 01:41 PDT
Search page Enter new bug
Bug#: 305134 alias:
Hardware: All
Reporter: Ben Goodger (use ber
org for email)
<bugs@bengoodger.c
Product: Firefox
OS: All
Add CC:
Component: RSS Discovery and Preview
Version: unspecified
CC: alex@spamcop.net
axel@pike.org
bugs.mano@sent.com
bugtrap@psychoticwol
bugzilla@dougweb.org
Remove selected
Status: RESOLVED
Priority: –
Resolution: FIXED
Severity: normal
Assigned To: Ben Goodger (use ben at mozilla dot org for email) <bugs@bengoodger.com>
Target Milestone: —
QA Contact: nobody@mozilla.org
URL:
Summary: Remove FeedView from Firefox 1.5
Status Whiteboard:
Flags: (Help!)
mtschrep: blocking1.8b4rc
bugs: blocking1.8b4
blocking1.9a1
blocking-aviary1.0.7

Many issues/bugs



JIRA Atlassian Support System

HOME | BROWSE PROJECT | FIND ISSUES | CREATE NEW ISSUE | ADMINISTRATION

Atlassian Support System

Issues: Ready for work (Displaying 15 of 45)
Oldest first

CSP-3924	Gallery macro opening images at full size is inconvenient for large images
CSP-3491	slow editing with confluence
CSP-3940	confluence seems to be leaking database connections
CSP-3942	Page Layout created in a Space becomes default for Site and all Spaces within
CSP-3950	Confluence 2.2 backup failed
CSP-3921	Cannot get RSS feed to work in confluence
CSP-3959	Userdata in embedded db is inconsistent - useres could not be removed
CSP-3935	Illegal State Exception
CSP-3969	Import into Confluence 2.1/Oracle 10g failed
CSP-3970	Confluence crash when viewing Info on page
CSP-3972	Delivery Status Notification (Failure)
CSP-3914	Unable to connect to a SQL Server db with named instance
CSP-3882	Windows AD LDAP authentication issues - escaping comma?
CSP-3894	"Title missing" message on save, even though title exists
CSP-3978	Custom Page Layout is lost with 2.2 upgrade

Issues: Ready to close (Displaying 1
Oldest first

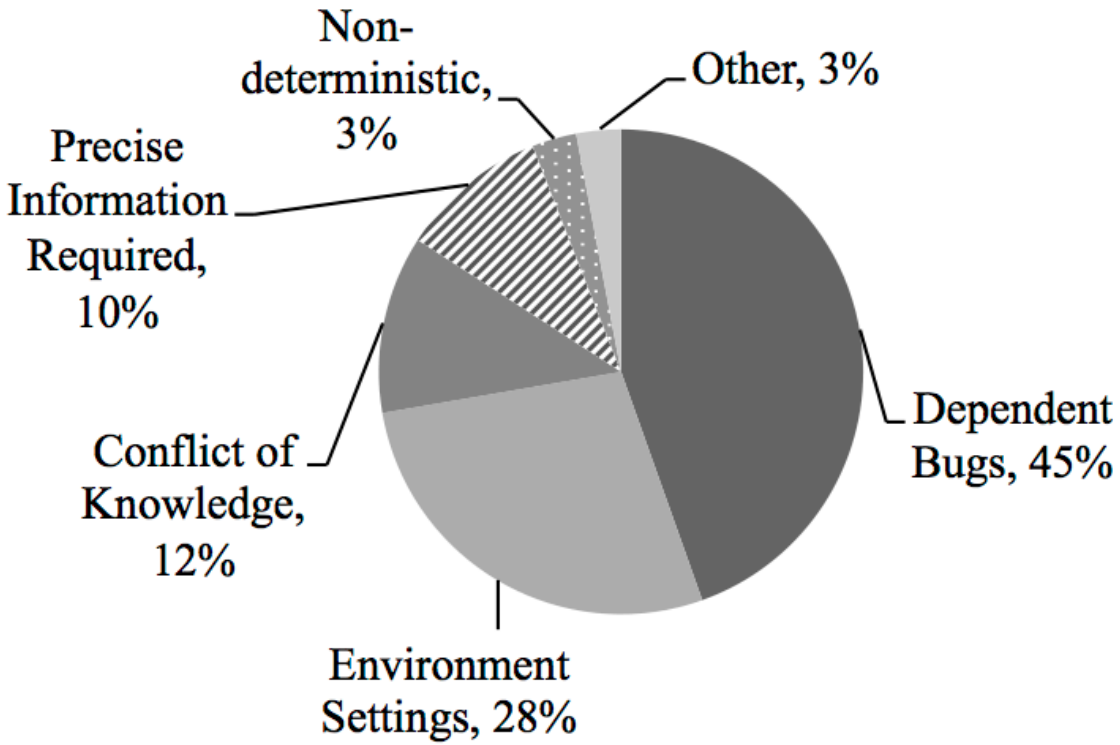
CSP-2866	Confluence 2.0 backu
CSP-2072	Database configurati
CSP-2641	Import pages from di
CSP-2647	RSS Generator
CSP-2669	'Search' tab on 'Creat
CSP-2615	Difficulty setting up n
CSP-2640	Startup problems
CSP-2612	Problems using Jira
CSP-2631	Implementing LDAP
CSP-2553	DWREngine undefin
CSP-2548	Graphiz-Macro stops
CSP-2695	Crashes 1-2 week
CSP-2736	I was trying to get a li
CSP-2728	browsing comment c
CSP-2701	Exception when tryin

How to reproduce the failure

- Problem Environment
  - Where failure was encountered, type of machine, OS, app version, etc
  - Hard to get exact environment, so we typically settle with *local environment*
- Program execution
  - User inputs, actions, time, etc.
- Make program run and fail deterministically given the specific conditions

Most annoying bug reports?

- Non-reproducible bugs
  - Difficult (or impossible?) to reproduce



Software Migration

Migrate your app

- From desktop to web
- From iOS to Android
- From RPC to SOA
- From Java to JavaScript

Software Reverse Engineering

"Reverse engineering is the process of analyzing a subject system to create representations of the system at a higher level of abstraction."

- Can be manual or automated
  - E.g. generate a basic UML class diagram
  - E.g. generate program traces and build sequence diagrams

Reverse Engineering Tools

- Quite a few of them are commercial…
  - **Rational Rose**
- Java decompilers (produce source code from binary)
  - JAD, Legal issues with reverse engineering!
- Standalone:
  - ArgoUML http://argouml.tigris.org/features.html
  - Crawljax: crawls and generates a state-flow graph automatically http://crawljax.com
  - Clematis: Reverse engineer a model/visualization from the application https://github.com/saltlab/clematis

Software Maintenance Subfields

- Debugging
- Reverse engineering
- Re-engineering, migration
- **Program Comprehension**

Program Comprehension

- During maintenance:
  - programmers **study the code** about **1.5** times longer than any documentation
  - programmers spend much more time **reading the code** than editing it
- Experts follow **dependency links**
  - …while novices read *sequentially*
- Much knowledge comes from outside the code
  - implicit relations between objects/functions

Program Comprehension

- Make implicit relations more explicit
- IDE plugins
- Software visualization often helps

Example: Clematis

- **What**: Helps developers understand event-driven interactions in JavaScript programs
- **How**: Instruments the code, traces at runtime, builds models, and visualizes!
- **Results**: 60% more accurate and 40% faster developers

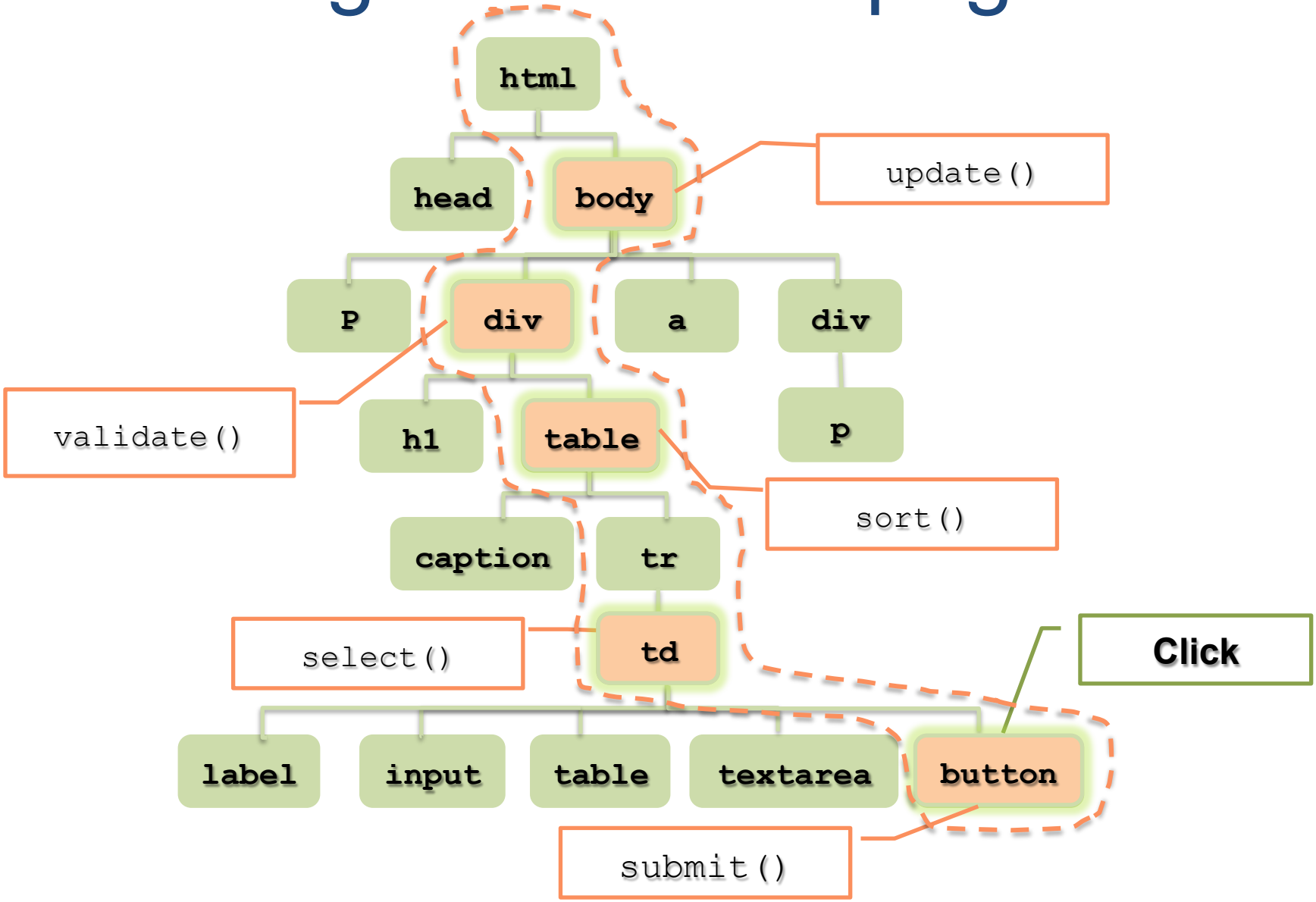
Challenges

- JavaScript
  - Event driven
  - Dynamic
  - Asynchronous
- Understanding the dynamic behavior and the control flow
  - Lower-level events
  - Their interactions

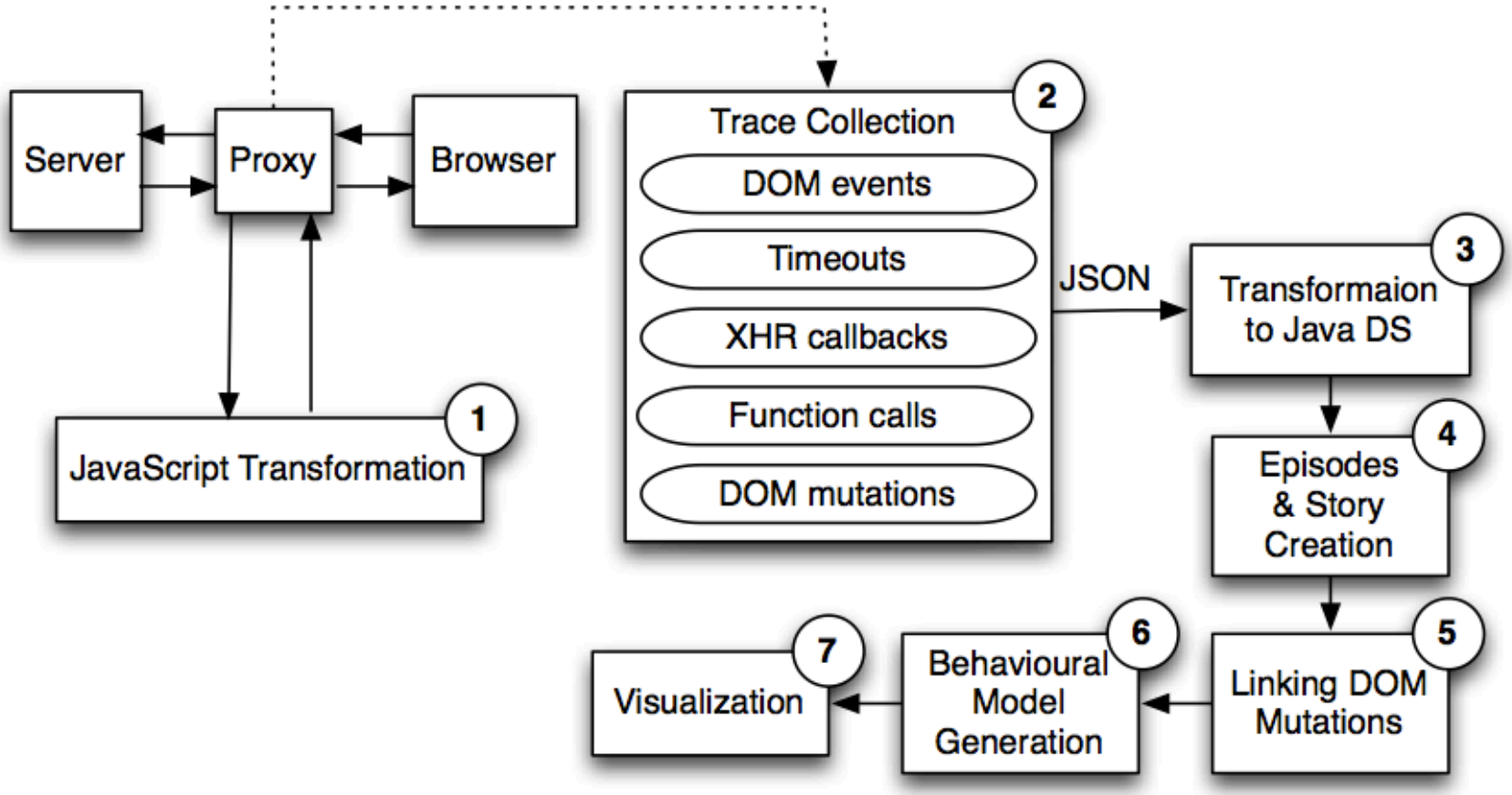
Challenge: Event Propagation

```
$(`#button’).click(submit);
 …
function submit() {
  $.get(‘/register/' , { name } , function(data) {
    $(‘#regMsg’).append(data);
});
…
```

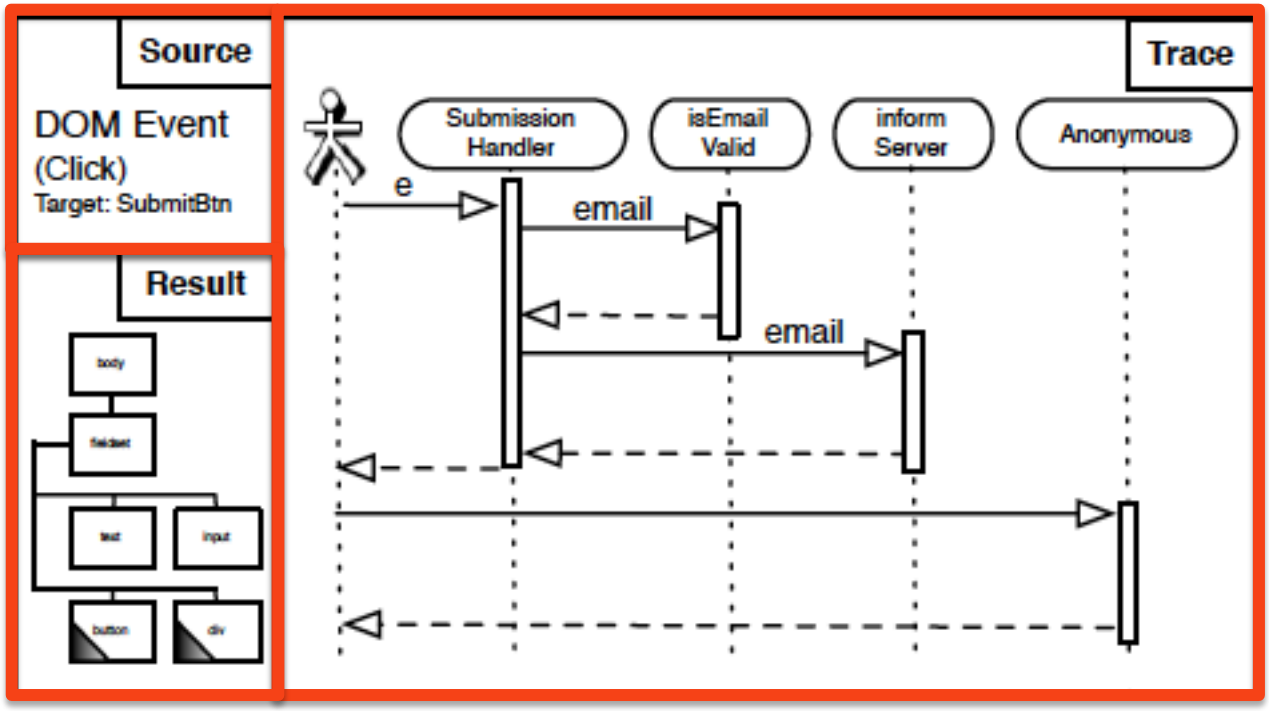
Challenge: Event Propagation



Approach

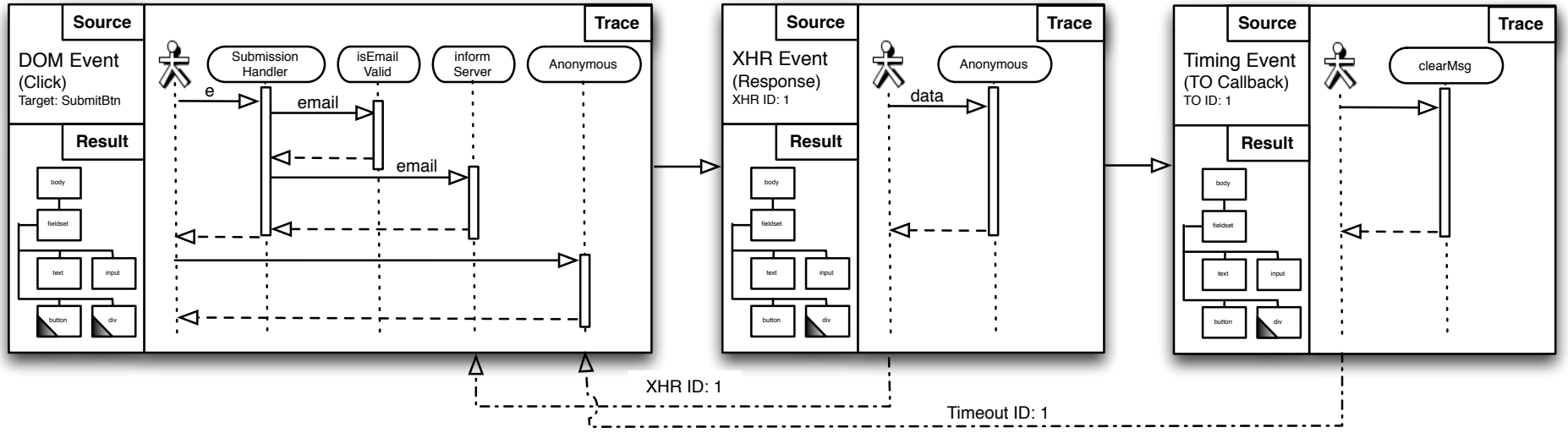

Server
Proxy
Browser
1
JavaScript Transformation
2
Trace Collection
DOM events
Timeouts
XHR callbacks
Function calls
DOM mutations
JSON
3
Transformaion to Java DS
4
Episodes & Story Creation
5
Linking DOM Mutations
6
Behavioural Model Generation
7
Visualization

Model: Episodes

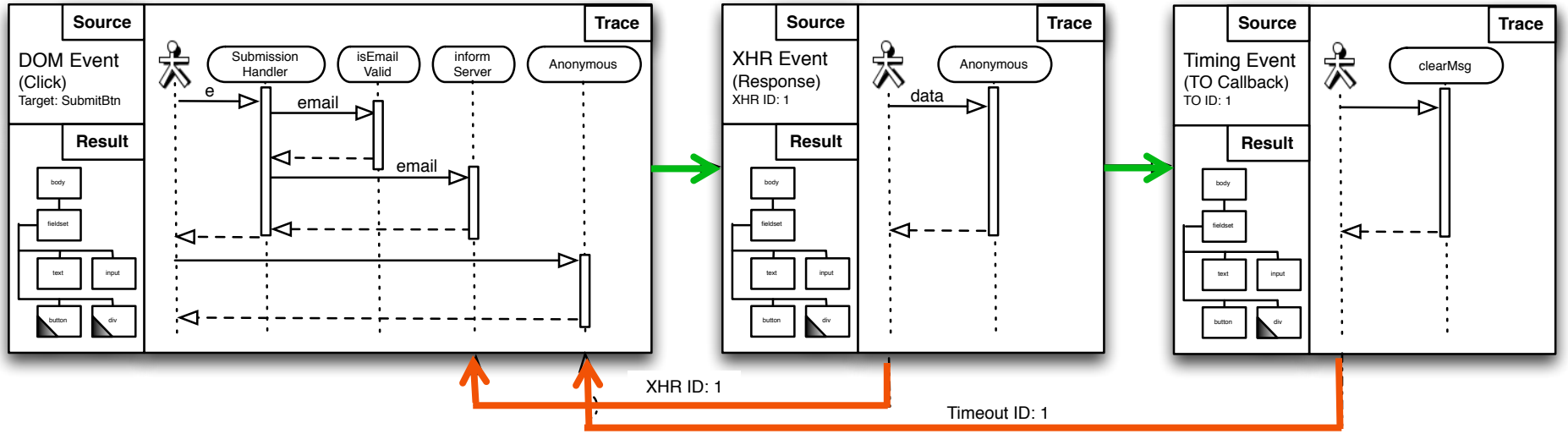


- A period of JavaScript execution
- Start and end points

Model: Story


Source
DOM Event
(Click)
Target: SubmitBtn
Result
body
fieldset
text
input
button
div
Trace
Submission Handler
isEmail Valid
inform Server
Anonymous
e
email
email
Source
XHR Event
(Response)
XHR ID: 1
Result
Trace
Anonymous
data
Source
Timing Event
(TO Callback)
TO ID: 1
Result
Trace
clearMsg
XHR ID: 1
Timeout ID: 1

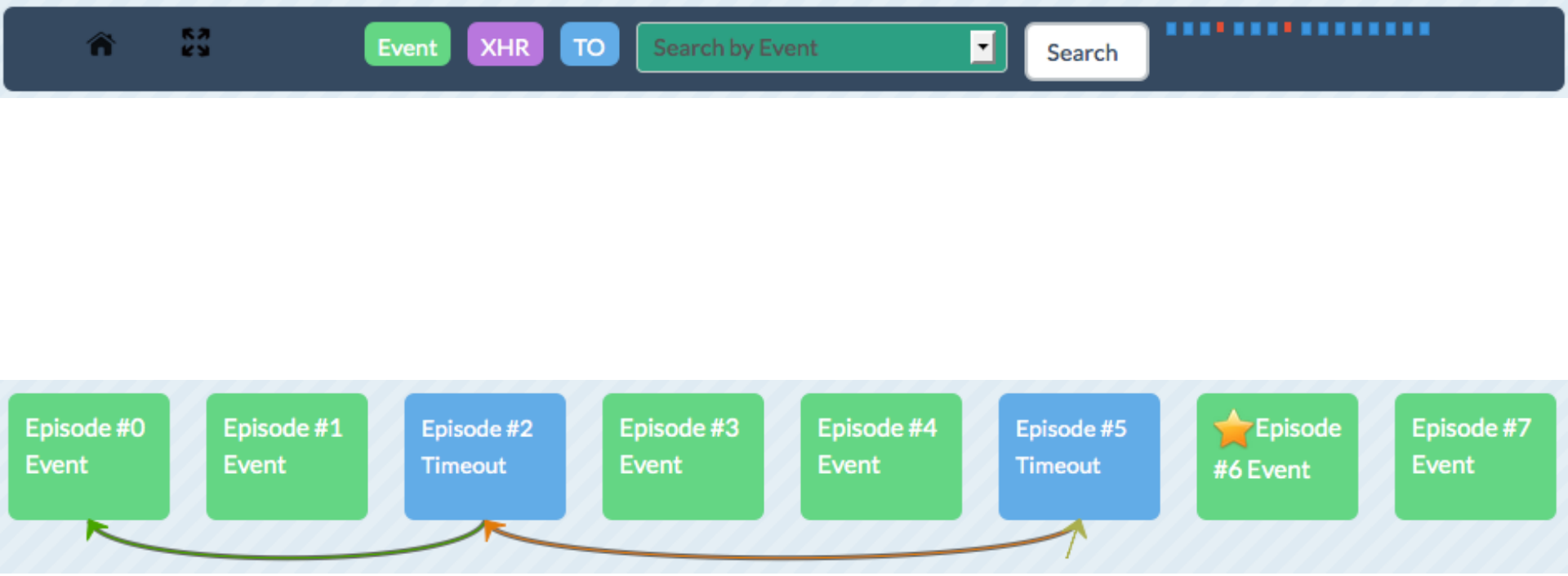
Model: Links



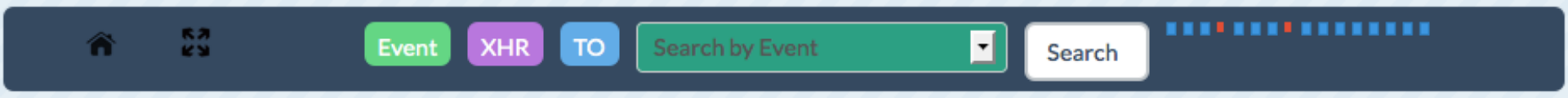
Temporal

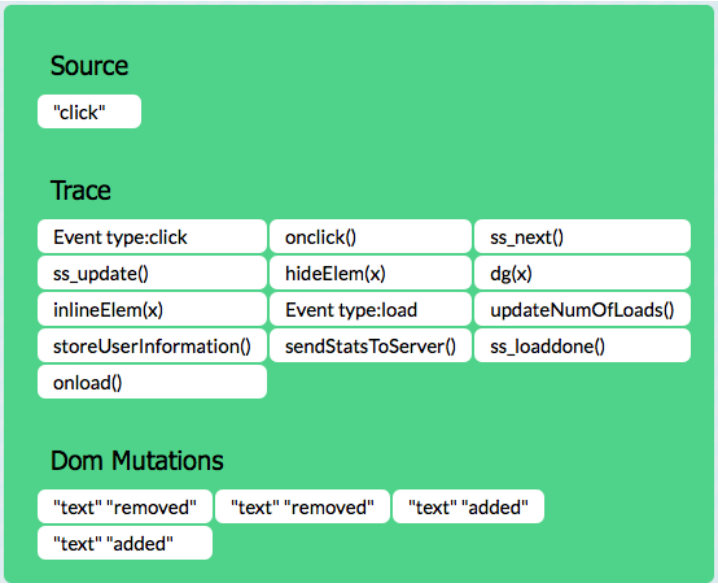
Causal

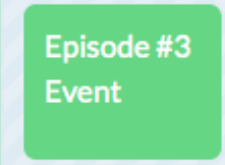
Visualization: Overview

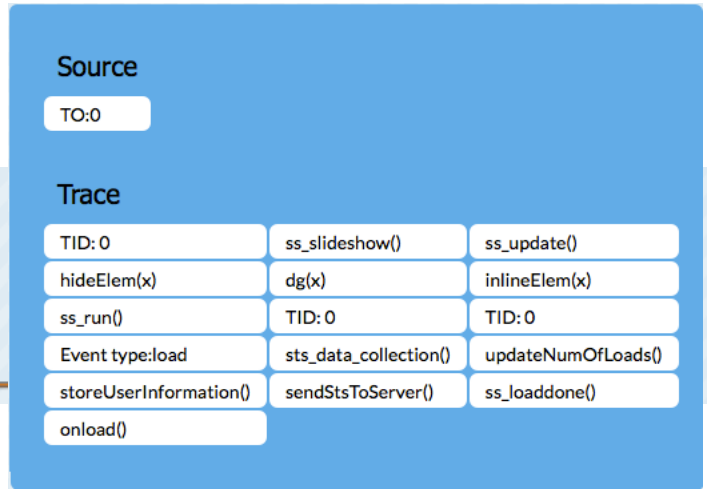


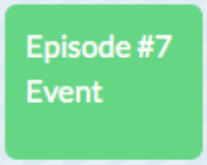
Visualization: Zoom Level 1

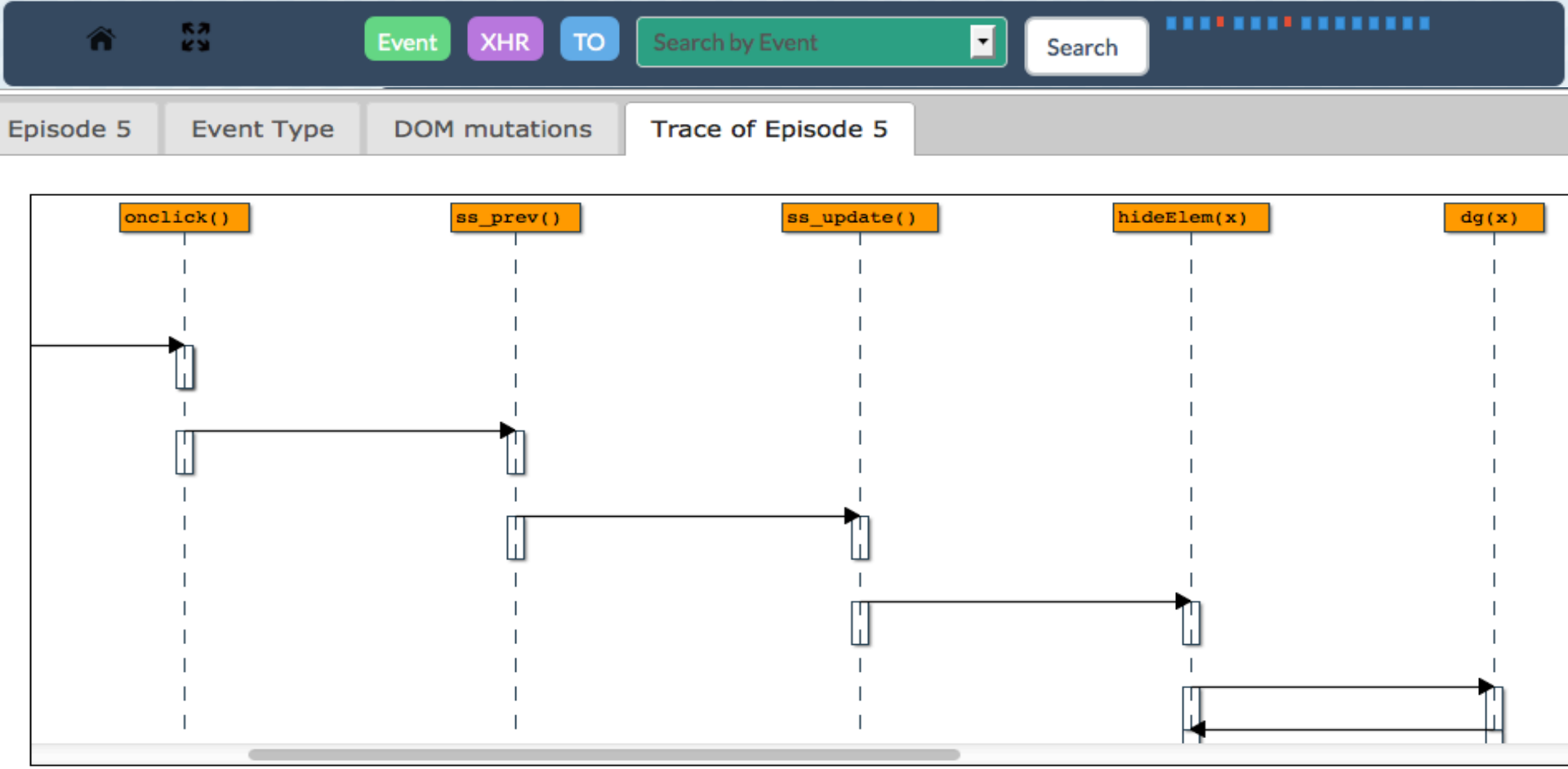












phorm.js

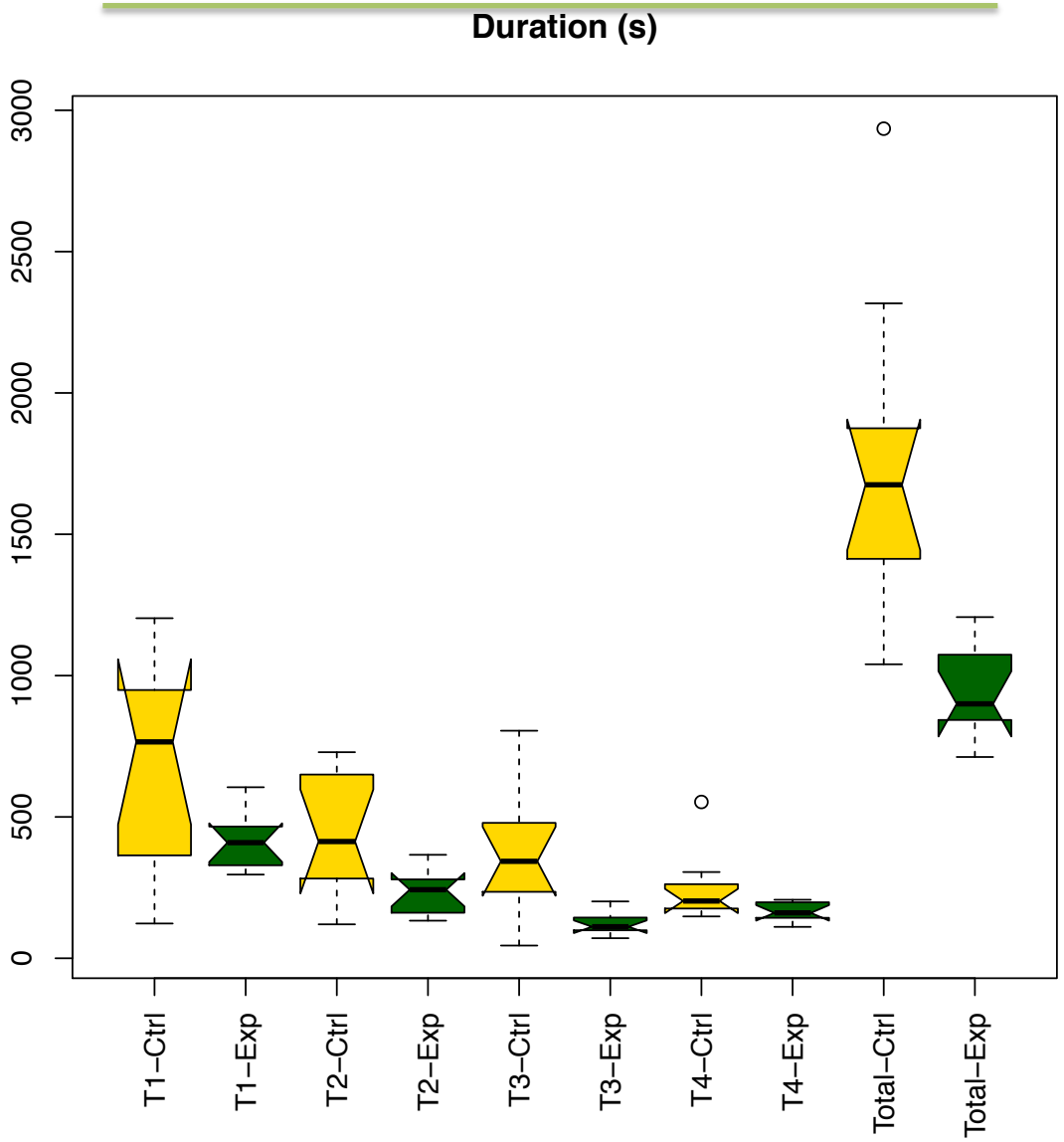
```
function ss_update() {
        ss_cur = Math.max(ss_cur, 0);

        if (ss_cur >= ss_date.length) {
                hideElem('ss_link2');
                showElem('ss_theend');
                ss_cur = ss_date.length;
                var a = dg('ss_n');
                a.innerHTML = "Final";
                if (ss_play)
                        ss_playpause();
        }
```

Visualization: Zoom Level 2

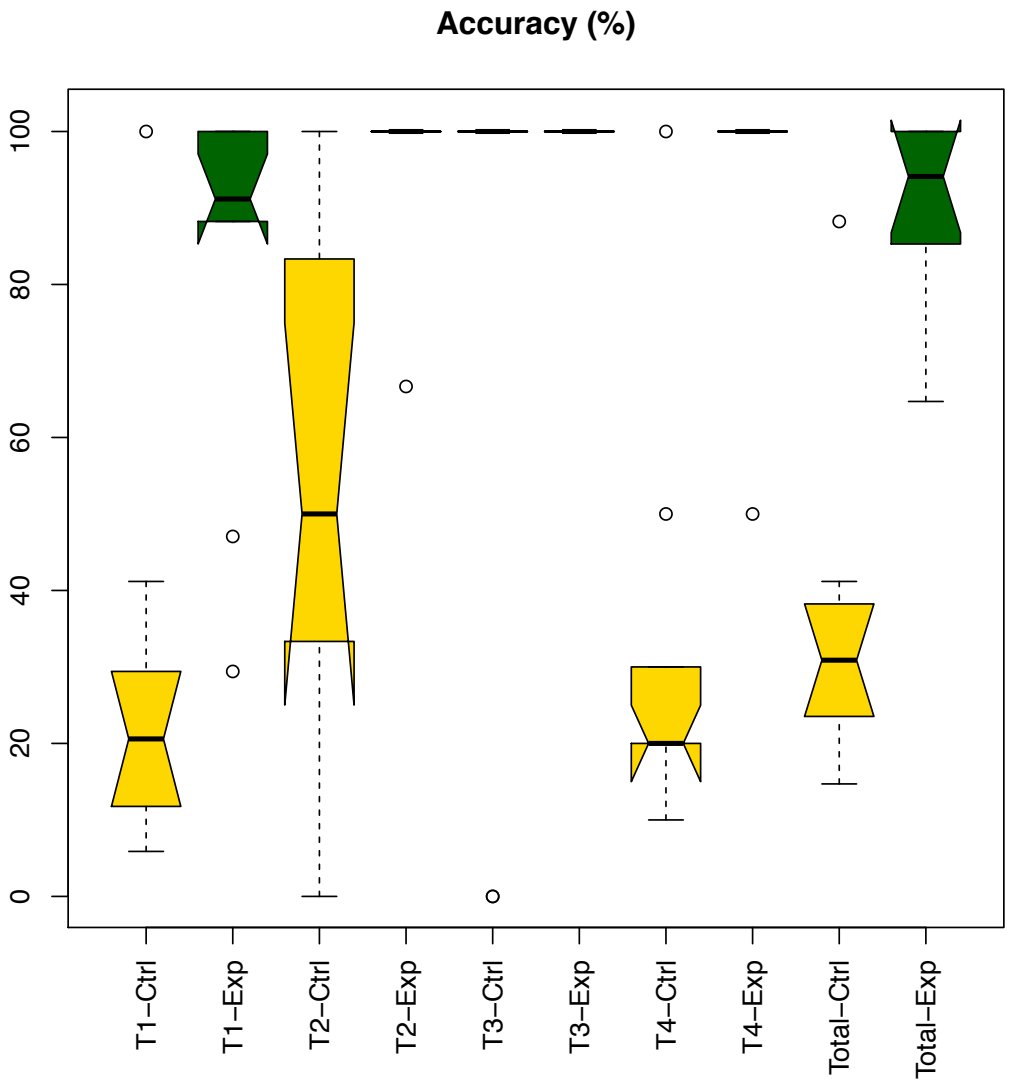
Duration

Task	Clematis		Other	
All	15:37	<<	29:12	(47%↑)



Accuracy

Task	Clematis		Other	
All	90	>>	35	(61%↑)



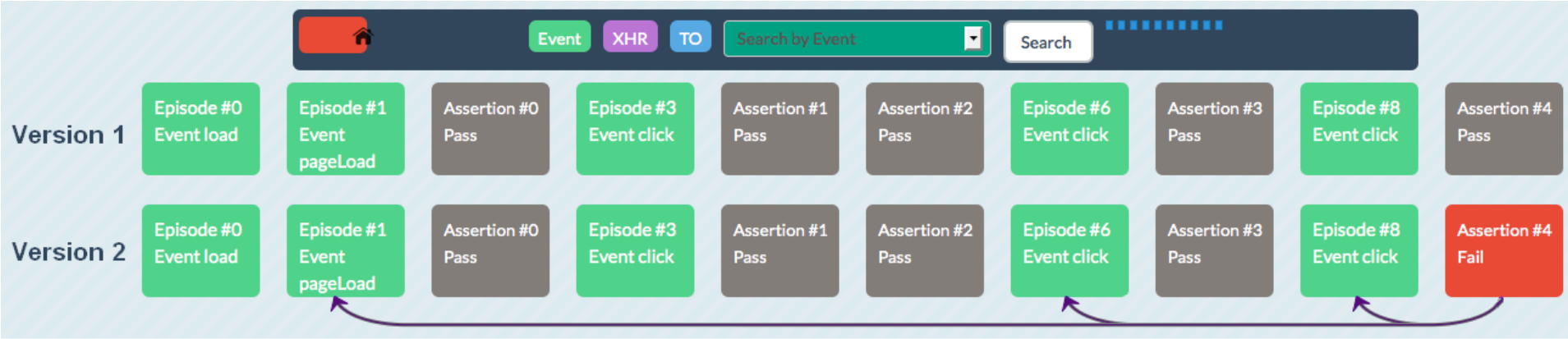
Visualizing Test Failures

```
@Test
public void testSlideShow() throws Exception {
    driver.get("http://localhost/?feat=slideshow");

    // Test 'Next' button
    driver.findElement(By.linkText("Next")).click();
    assertEquals("2", driver.findElement(By.id("pictureNumber")).getText());

    // Test 'Previous' button
    driver.findElement(By.linkText("Previous")).click();
    assertEquals("1", driver.findElement(By.id("pictureNumber")).getText());
}
```

Visualization: assertion failure



In-class exercise: Maintenance issues

In your lab groups,

You have been given the task of discovering the factors that affect the **maintainability** of the systems developed by your company.

1. How would you determine appropriate maintainability processes and metrics for the company?

Useful references

- Software Maintenance and Evolution: A Roadmap
- Michael Feathers, "Working Effectively with Legacy Code" http://www.amazon.com/exec/obidos/ISBN=0131177052/
- L. A. Belady and M. M. Lehman, "A model of large program development," IBM Systems Journal, vol. 3, pp. 225-252, 1976.