

Lecture 20

Evolving the Requirements

Com S/SE 409/509

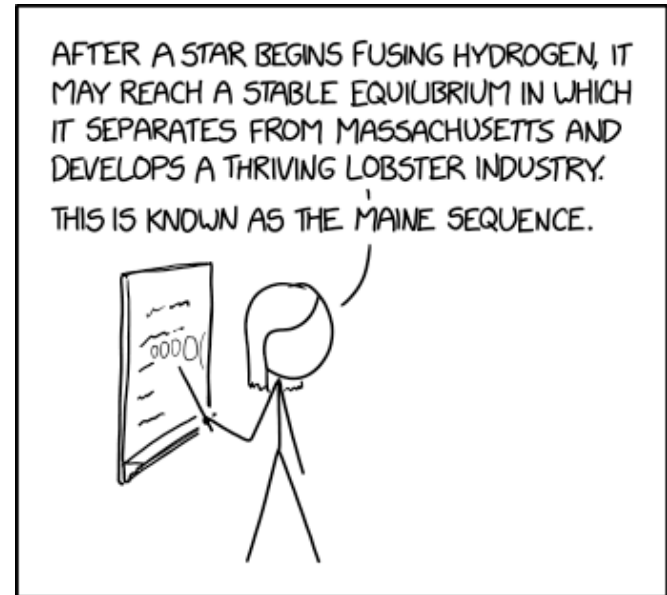
Robyn Lutz

rlutz@iastate.edu

Robyn's Office Hours: **Tues & Thurs, 9:30-10:45**

Wandi's Office Hours: **Mon 10 am. & Wed 7 pm**

Olukorede's Office Hours--**Wed 10**



Stellar Evolution

xkcd.com/license.html

Requirements evolution

Lecture based on: [van Lamsweerde, Requirements Engineering, 2009]

His summary: ***“Prepare for requirements to change”***

Requirements change for a variety of reasons, both before & after software is deployed

Requirements changes create a **requirements management** problem: the SRS & perhaps the design, code, users' manual, etc. may have to change

Requirements evolution = Requirements change

Requirements will change

Prepare for it

Manage it

Basic types of requirements changes:

Add a software requirement

Delete a software requirement

Modify a software requirement

Requirements change management

Anticipate

- identify likely requirements changes

- make sure your software can readily adapt to support them

Evaluate

Agree

Propagate

Good traceability helps you manage requirements change

Traceability links are mappings between items

- traces backwards to where the requirement item came from

- traces forwards to where the requirement item will be used

Ex: the product use case: “produce de-icing schedule”

- traces backwards** to context diagram arrow: “road de-icing schedule”

- trace forwards** to functional requirements describing steps:

 - “... the product shall accept a valid district identifier ...

 - the product shall determine when each road section will freeze.”

Good traceability helps you manage requirements change, cont.

Traceability helps you understand what's changing

Traceability helps you localize impact of the change

When requirements change, you need to **update the traceability links**

Effects of requirements changes

1. Creates versions (also called revisions)

Requirements evolve to fix &/or improve the current version of the product over time

2. Creates variants

Requirements evolve to create more products in a product line, such as for different customers with differing requirements

Adds/deletes features

Check your understanding: preparing for change

Identify 8 likely changes to EverGreen's software requirements in its first 2 years after deployment, due to each of these causes [van Lamsweerde, 2009]

1. Error found in the SRS:

Example: Was: "At StartUp the software shall be in manual mode."

Corrected: "At StartUp the software shall be in automatic mode."

2. Better understanding of customer needs

3. New feature (functional)

4. Improved quality needed (nonfunctional)

5. New type of users

6. New legal regulation

7. Technology change

8. New business opportunity