Requirement Management, Agile RE

Lecture 8, DAT230, Requirements Engineering
Robert Feldt, 2012-10-02

Recap

- Req Validation = Activities to ensure requirements are the right ones, have high quality etc
- Reviews are the main technique
  - Fagan Inspections (IBM) = most formal
  - Selective Homeworless = in between
  - Ad hoc reviews = least formal, most common
- Prototyping can also help pinpoint problems
- Research shows reviews are cost-effective but industry want more focused reviews & checklists

Change Management

- Requirements baselining
- Procedures for new and changed requirements
  - How to propose
  - How to process
  - How to negotiate
  - How to communicate
- Impact analysis procedures
- How changes are reflected in project plans & commitments

Three main reasons for change

- Changing market demands
- Developers increased understanding
- Organizational reasons - strategy change, scope reduction

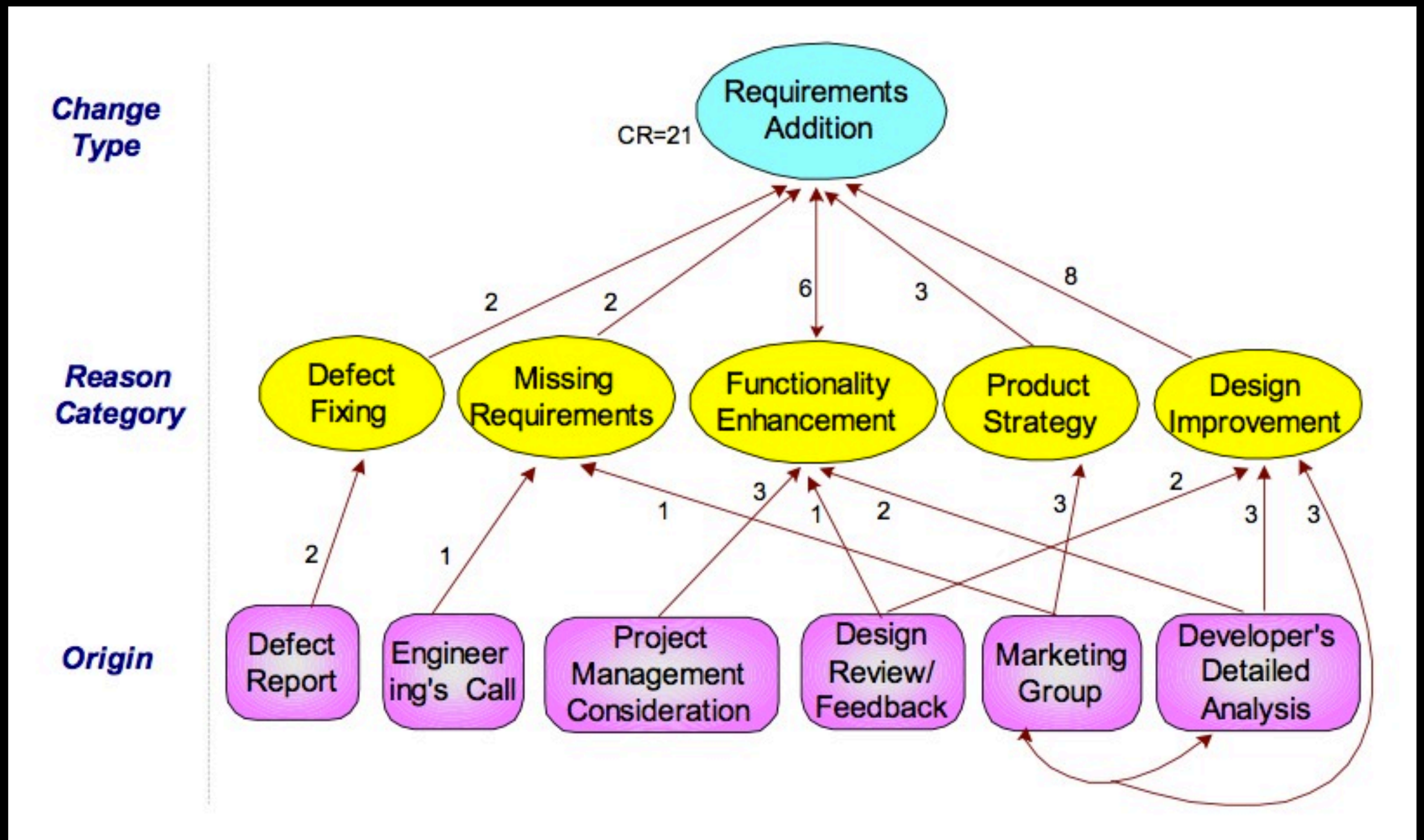
[Nurmulian2004]

Taxonomy of Req Change

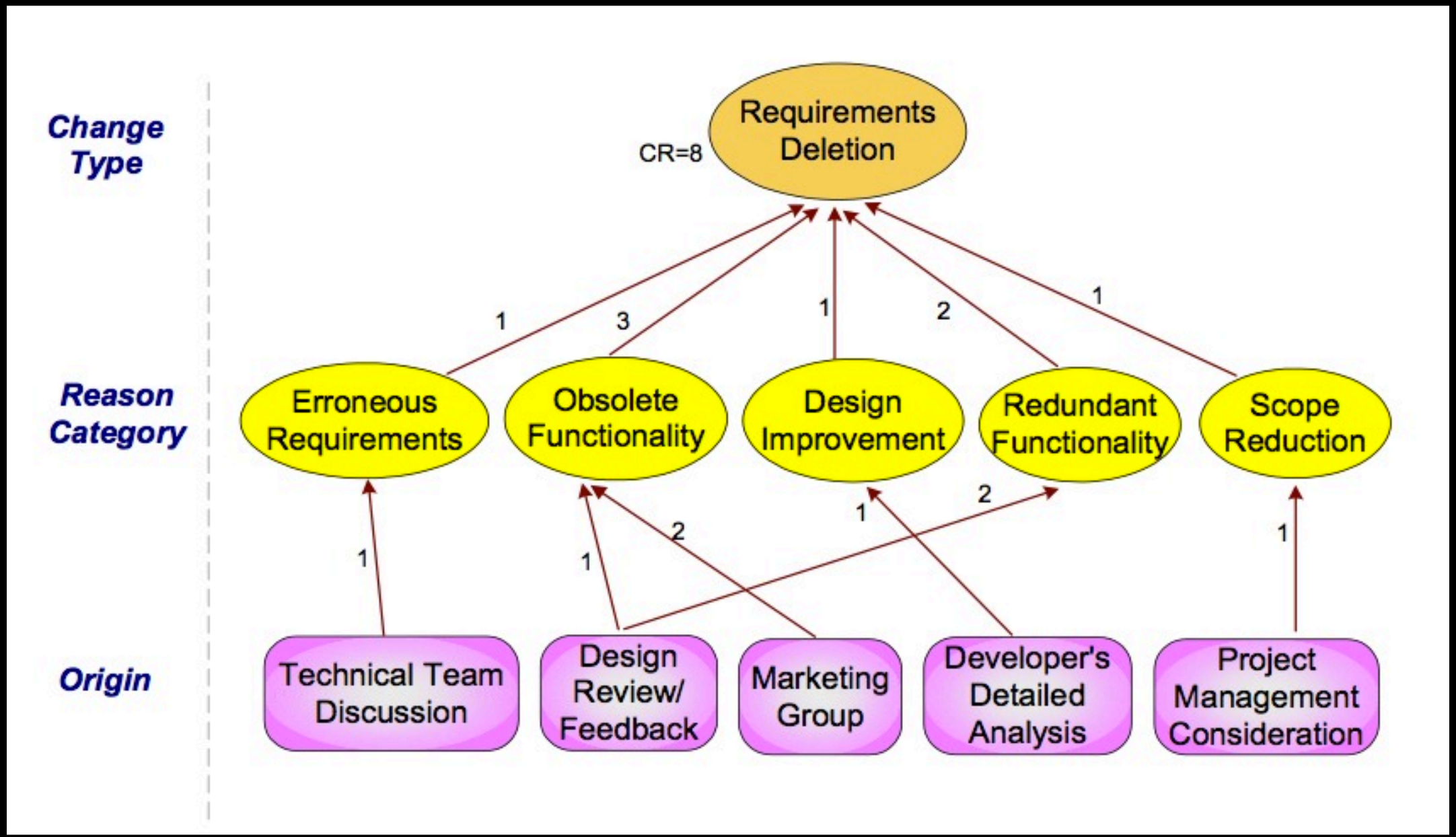
- Type: Add, Delete, Modify
- Reason: Defect fix, Func. Enhancement, Design improvement, ...
- Origin: Defect report, Marketing group, Review, ...

[Nurmuliani2004]

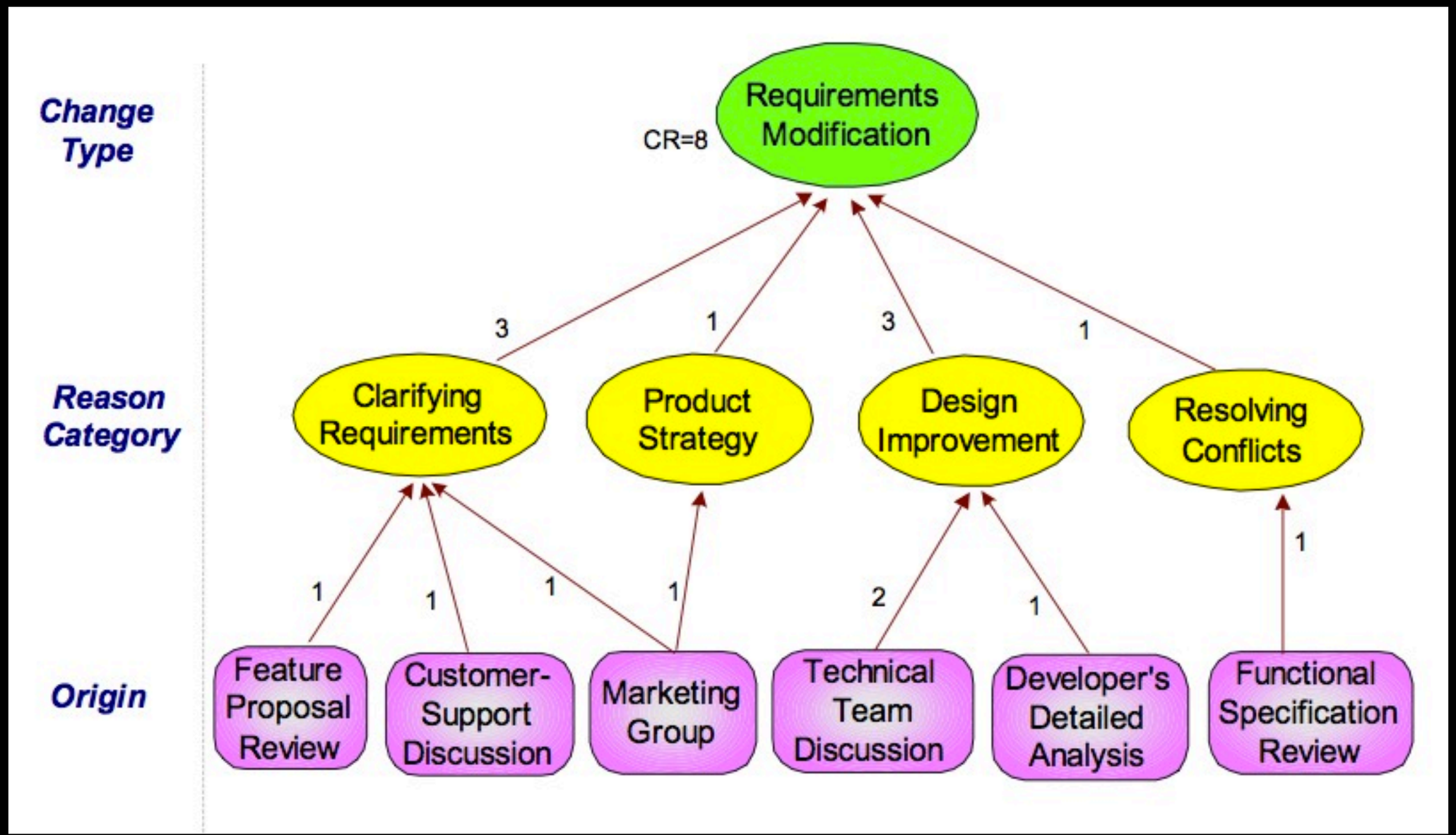
Requirement Additions



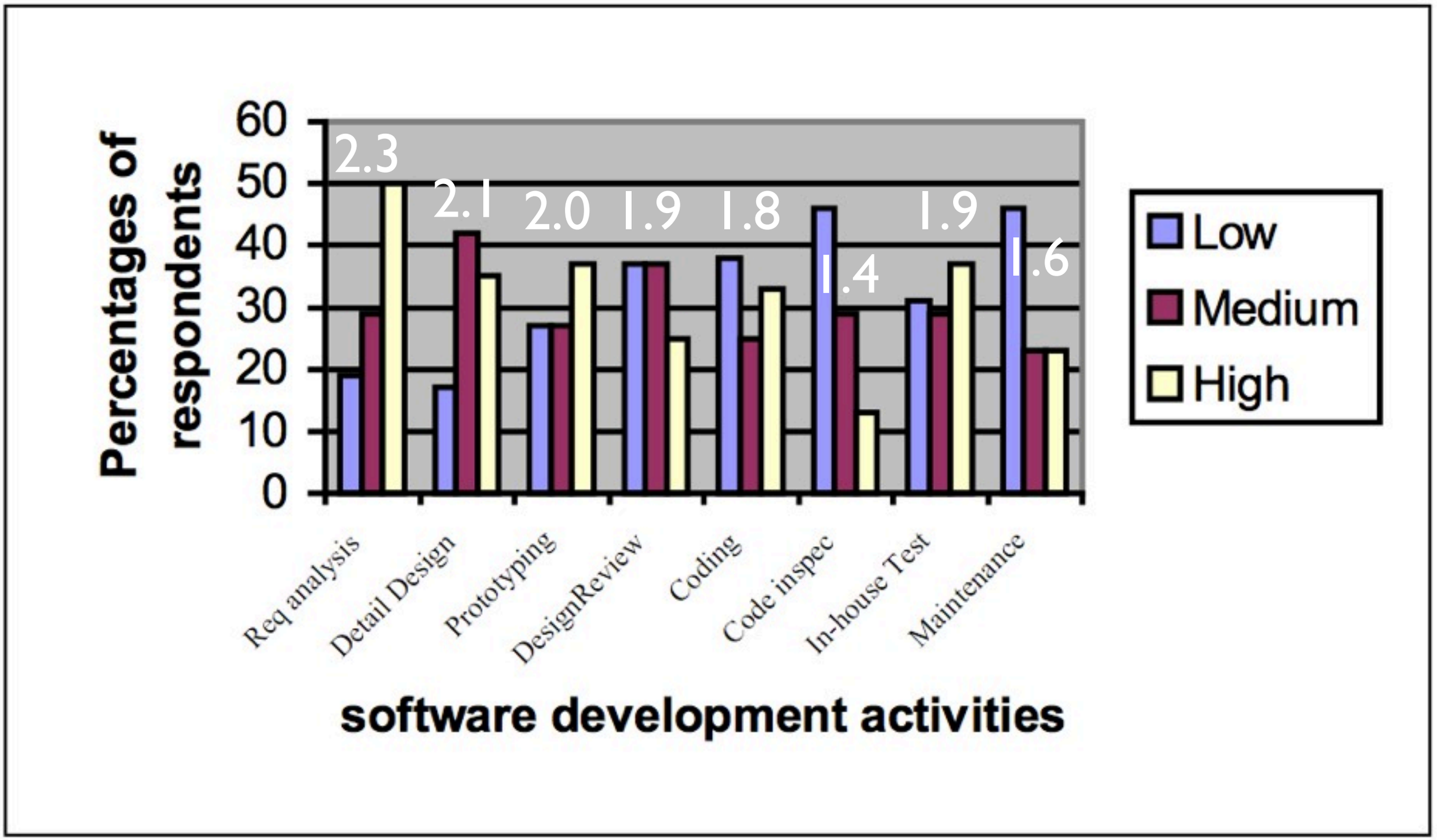
Requirement Deletions



Requirement Modifications



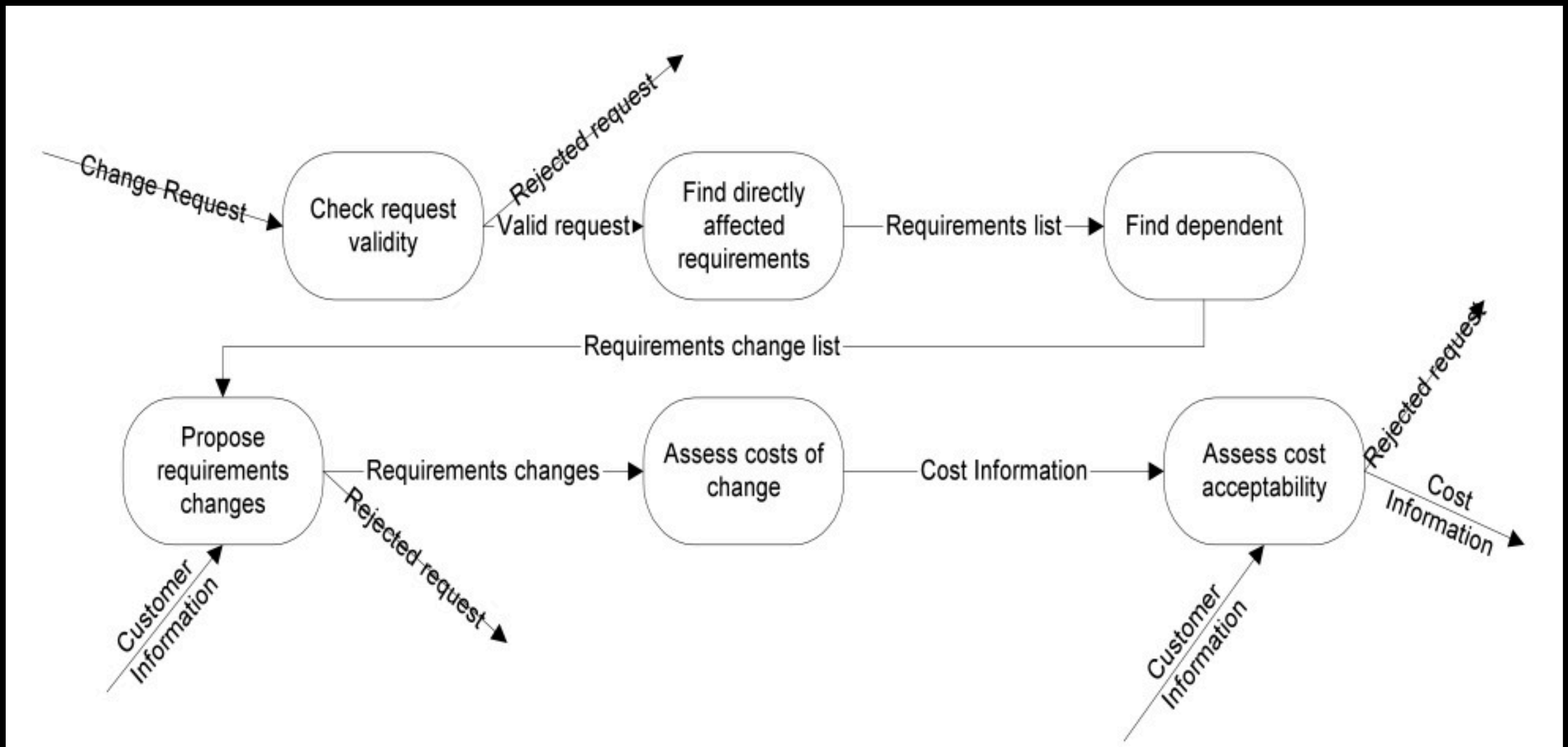
When are the changes?



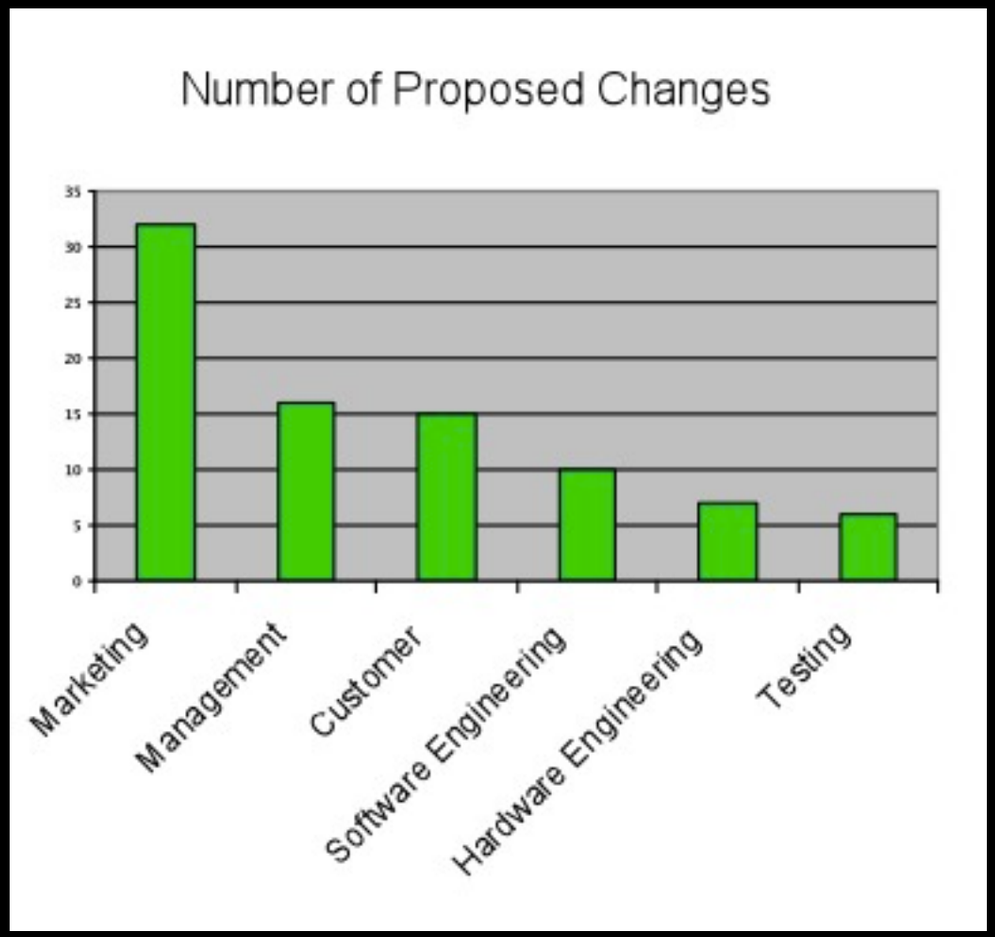
Preparing for change

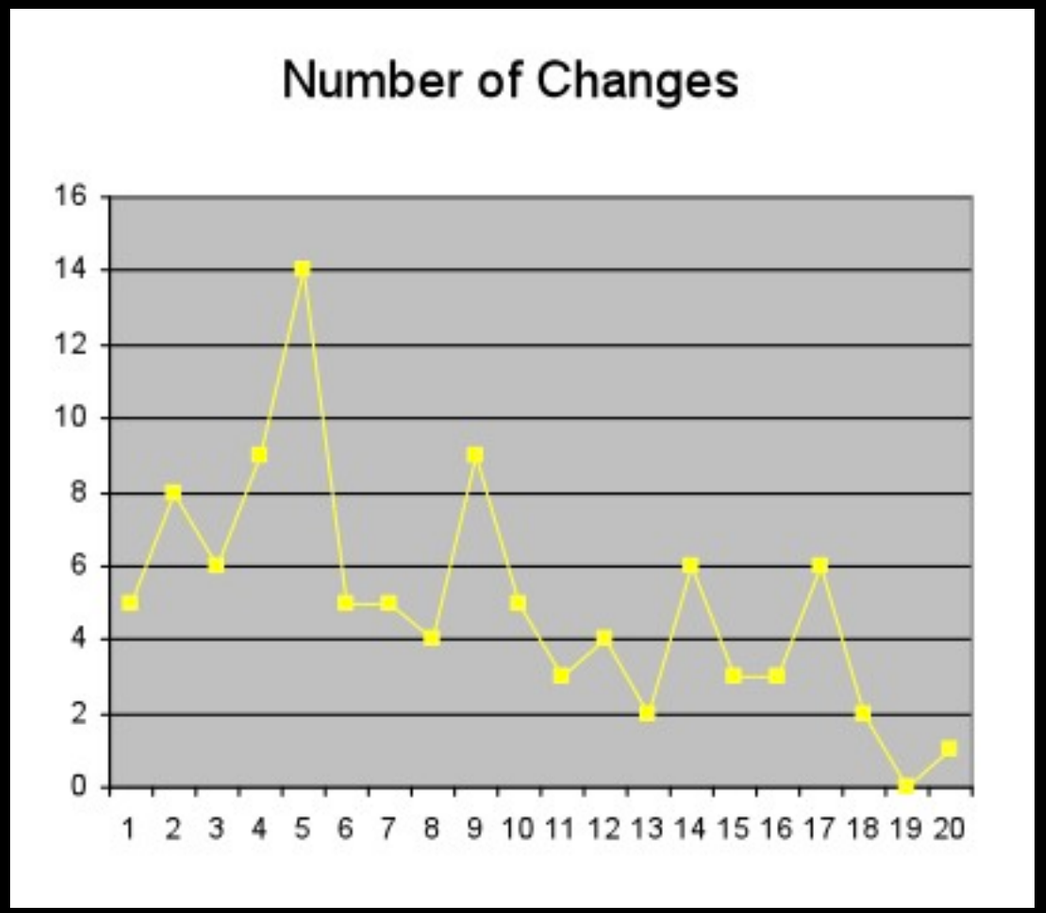
- Accept that changes are inevitable
- Identify Volatile Requirements
- Establish single channel for change requests
- Manage change hierarchically

Change Management Process



Measuring Change Activity





- Assess stability of requirements
- Identify improvement opportunities
- Alerts to project planning

What is “Agile Development Processes”?

ag·ile

[**aj**-*uh*l, -ahyl] Show IPA

–adjective

1. quick and well-coordinated in movement; lithe: *an agile leap.*
2. active; lively: *an agile person.*
3. marked by an ability to think quickly; mentally acute or aware: *She's 95 and still very agile.*

Use **agile** in a Sentence

Origin:
1570–80; earlier *agil* < Latin *agilis,* equivalent to *ag-* (base of *agere* to do) + *-ilis* -ile

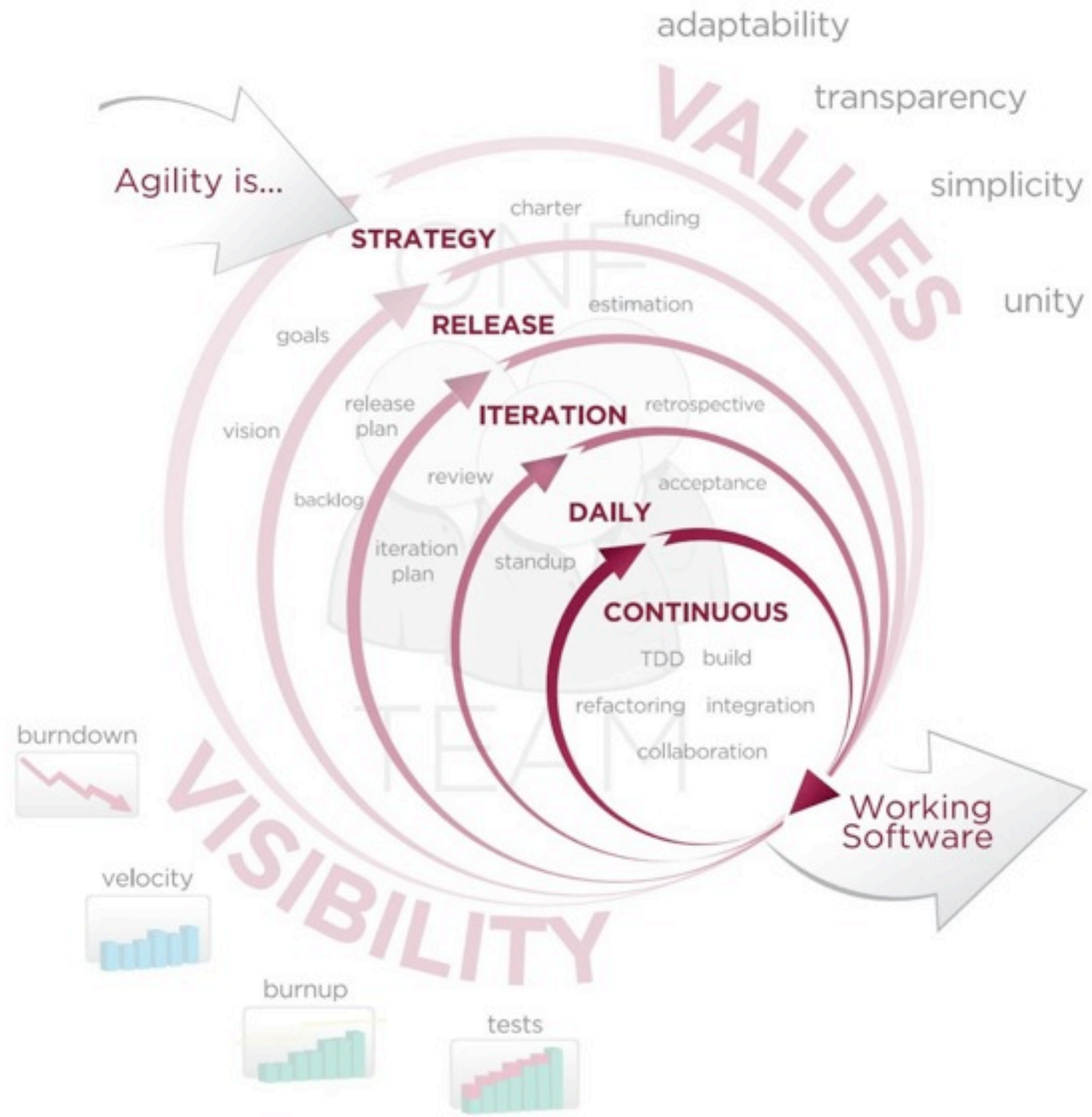
—Related forms
ag·ile·ly, *adverb*
ag·ile·ness, *noun*
un·ag·ile, *adjective*
un·ag·ile·ly, *adverb*

—Synonyms
1. nimble, sprightly. 2. brisk, spry.

—Antonyms
1. awkward. 2. sluggish, lethargic.

[Dictionary.com 2011]

AGILE DEVELOPMENT



ACCELERATE DELIVERY

Key RE aspects of agile (Schwaber)

- **Iterative development -** frequent iterations generate increments of work (inspected, baselined)
- **Increments of work -** working system/functionality instead of documents (feedback based on real value)
- **Collaboration -** customers and developers together
- **Daily meetings -** daily status & feedback
- **Adaptation -** change org daily to best deliver quality
- **Emergence -** system, architecture & reqs emerge during project

“Agile” RE in practice

- [Cao2008]
- Interviews with 54 personer in 16 companies
  - All used XP or SCRUM, fully or partially
- Questions:
  - How does “agile” developers work with RE?
  - Which advantages and disadvantages?

What do they do?

Agile requirements-engineering practices in 16 organizations

Adoption level	Practice						
	Face-to-face communication	Iterative RE	Extreme prioritization	Constant planning	Prototyping	Test-driven development	Reviews & tests
High	8	9	10	8	8	5	11
Medium	8	5	6	6	3	1	4
Low	0	2	0	2	0	0	1
None	0	0	0	0	5	10	0

Agile RE Practices - Pro / Con

Face2Face communication & User stories

Iterative req engineering

Agile RE Practices - Pro / Con

Face2Face communication & User stories

Saves time

Iterative req engineering

Agile RE Practices - Pro / Con

Face2Face communication & User stories

- Saves time
- Customer drives

Iterative req engineering

Agile RE Practices - Pro / Con

Face2Face communication & User stories

Saves time

Requires trust

Customer drives

Iterative req engineering

Agile RE Practices - Pro / Con

Face2Face communication & User stories

- Saves time
- Customer drives
- Requires trust
- Not all user groups represented

Iterative req engineering

Agile RE Practices - Pro / Con

Face2Face communication & User stories

- Saves time
- Requires trust
- Customer on site
- Customer drives
- Not all user groups represented

Iterative req engineering

Agile RE Practices - Pro / Con

Face2Face communication & User stories

- Saves time
- Requires trust
- Customer on site
- Customer drives
- Not all user groups represented

Iterative req engineering

- Clearer reqs

Agile RE Practices - Pro / Con

Face2Face communication & User stories

- Saves time
- Requires trust
- Customer on site
- Customer drives
- Not all user groups represented

Iterative req engineering

- Clearer reqs
- Customer relation

Agile RE Practices - Pro / Con

Face2Face communication & User stories

- Saves time
- Customer drives
- Requires trust
- Not all user groups represented
- Customer on site

Iterative req engineering

- Clearer reqs
- Customer relation
- Minimal docs when problems

Agile RE Practices - Pro / Con

Face2Face communication & User stories

- Saves time
- Customer drives
- Requires trust
- Customer on site
- Not all user groups represented

Iterative req engineering

- Clearer reqs
- Customer relation
- Minimal docs when problems
- Cost & schedule estimation

Agile RE Practices - Pro / Con

Face2Face communication & User stories

- Saves time
- Customer drives
- Requires trust
- Customer on site
- Not all user groups represented

Iterative req engineering

- Clearer reqs
- Customer relation
- Minimal docs when problems
- Cost & schedule estimation
- Non-functional requirements

Agile RE Practices - Pro / Con

"Extreme" Prioritization

Managing Change through Constant replanning

Agile RE Practices - Pro / Con

"Extreme" Prioritization

Clearer reasons

Managing Change through Constant replanning

Agile RE Practices - Pro / Con

"Extreme" Prioritization

- Clearer reasons
- Re-prio is easier

Managing Change through Constant replanning

Agile RE Practices - Pro / Con

"Extreme" Prioritization

Clearer reasons

"Business Value" to narrow

Re-prio is easier

Managing Change through Constant replanning

Agile RE Practices - Pro / Con

"Extreme" Prioritization

Clearer reasons	"Business Value" to narrow
Re-prio is easier	Instable with re-prio

Managing Change through Constant replanning

Agile RE Practices - Pro / Con

“Extreme” Prioritization

- Clearer reasons
- “Business Value” to narrow
- Re-prio is easier
- Instable with re-prio

Managing Change through Constant replanning

- Fewer changes

Agile RE Practices - Pro / Con

“Extreme” Prioritization

- Clearer reasons
- “Business Value” to narrow
- Re-prio is easier
- Instable with re-prio

Managing Change through Constant replanning

- Fewer changes
- Smaller changes

Agile RE Practices - Pro / Con

“Extreme” Prioritization

- Clearer reasons
- Re-prio is easier
- “Business Value” to narrow
- Instable with re-prio

Managing Change through Constant replanning

- Fewer changes
- Smaller changes
- Architecture suffers

Agile RE Practices - Pro / Con

“Extreme” Prioritization

Clearer reasons	“Business Value” to narrow
Re-prio is easier	Instable with re-prio

Managing Change through Constant replanning

Fewer changes	Architecture suffers
Smaller changes	“Refactoring” not enough

Agile RE Practices - Pro / Con

Prototypes

Test-driven Development

Reviews & Acceptance tests

Agile RE Practices - Pro / Con

Prototypes

Quicker feedback

Test-driven Development

Reviews & Acceptance tests

Agile RE Practices - Pro / Con

Prototypes

Quicker feedback

Unrealistic dev speed expectations

Test-driven Development

Reviews & Acceptance tests

Agile RE Practices - Pro / Con

Prototypes

Quicker feedback

Unrealistic dev speed expectations

Test-driven Development

Tests capture reqs

Reviews & Acceptance tests

Agile RE Practices - Pro / Con

Prototypes

Quicker feedback

Unrealistic dev speed expectations

Test-driven Development

Tests capture reqs

Freedom to experiment

Reviews & Acceptance tests

Agile RE Practices - Pro / Con

Prototypes

Quicker feedback

Unrealistic dev speed expectations

Test-driven Development

Tests capture reqs

Requires close customer collab

Freedom to experiment

Reviews & Acceptance tests

Agile RE Practices - Pro / Con

Prototypes

Quicker feedback

Unrealistic dev speed expectations

Test-driven Development

Tests capture reqs

Requires close customer collab

Freedom to experiment

Developers resist

Reviews & Acceptance tests

Agile RE Practices - Pro / Con

Prototypes

Quicker feedback

Unrealistic dev speed expectations

Test-driven Development

Tests capture reqs

Requires close customer collab

Freedom to experiment

Developers resist

Reviews & Acceptance tests

Status report to customers

Agile RE Practices - Pro / Con

Prototypes

- Quicker feedback
- Unrealistic dev speed expectations

Test-driven Development

- Tests capture reqs
- Requires close customer collab
- Freedom to experiment
- Developers resist

Reviews & Acceptance tests

- Status report to customers
- Hard to create acc.tests