Requirement Management, Agile RE

Lecture 9, DAT230, Requirements Engineering
Robert Feldt, 2010-09-28

Notes about course

- Student course evaluators: We need 5 of you!
- Group assignment:
  - Doodle's on home page
  - FAQ will collect some common questions on home page
  - Don't miss deadlines late next week
  - Rubric for PostMortem will be uploaded this week

Recap from last lecture

Recap

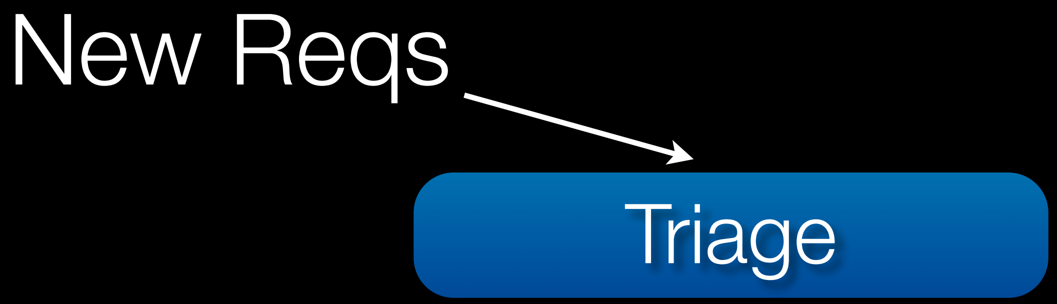
- Req traceability to follow links to and from reqs
  - to = sources, reasons, versions, releases
  - from = design, implementation, tests, use, refinement
- Need traceability for: Certification, Testing, Tracking, Changes
- Links: Satisfies, Dependency, Rationale, Evolves-to
- Prioritization = which into next release
- Common techniques: 100 dollar, yes-no, 5-way, cost-value
  - Triage (MDRE)

Triage (in MDRE)

Triage (in MDRE)

New Reqs

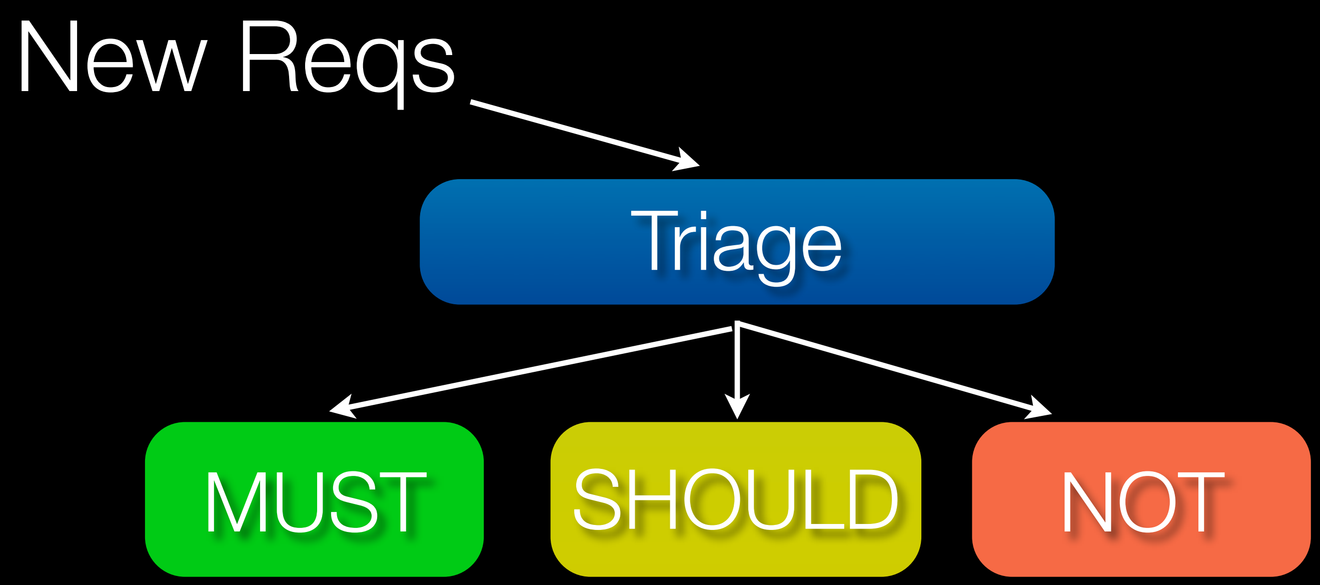
Triage (in MDRE)



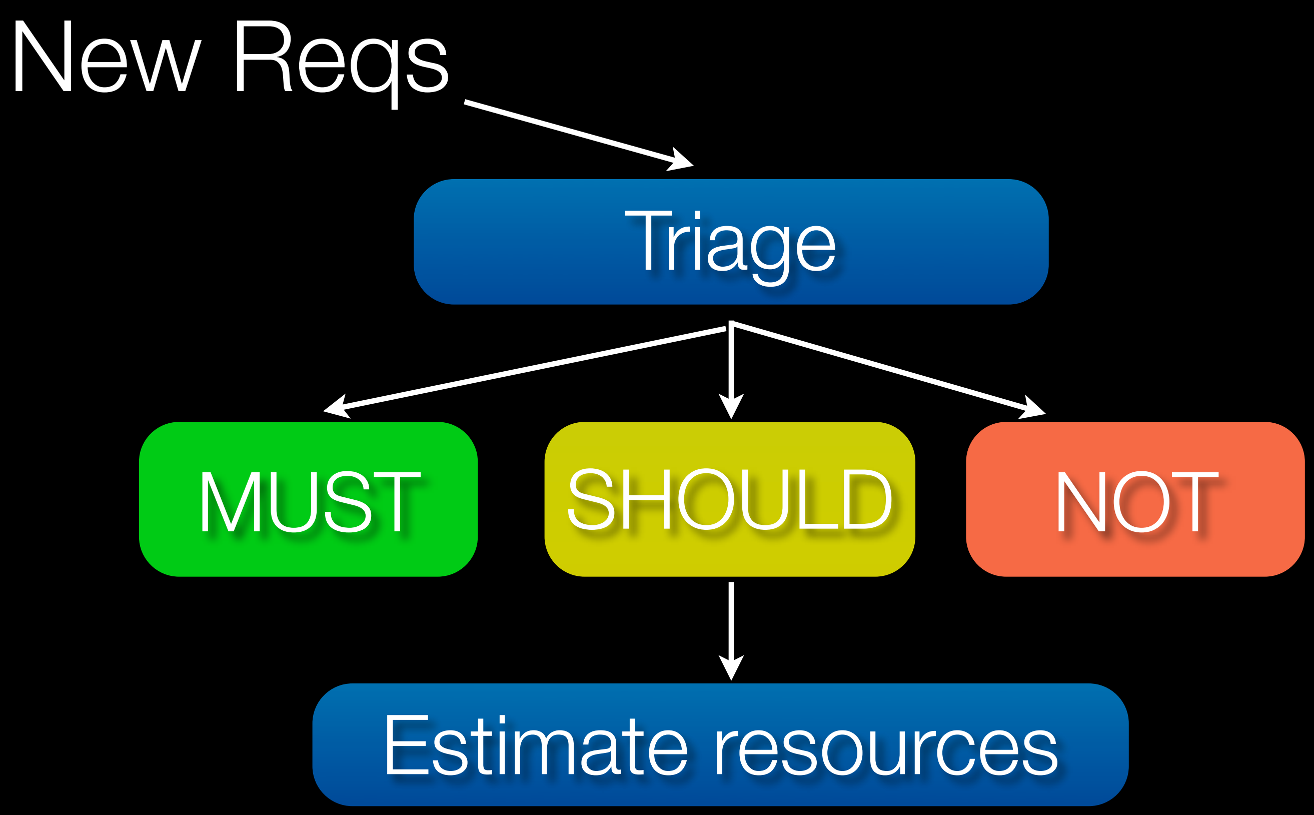
New Reqs

Triage

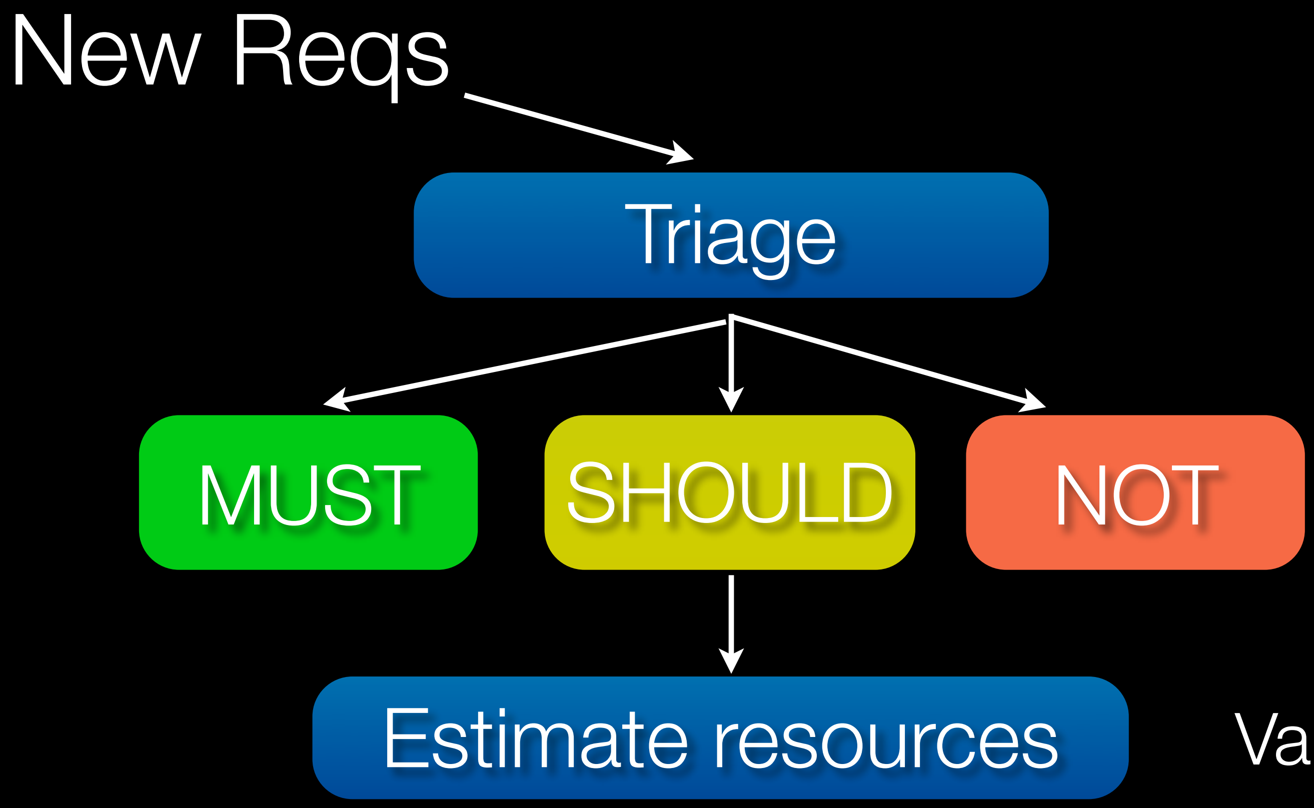
Triage (in MDRE)



Triage (in MDRE)

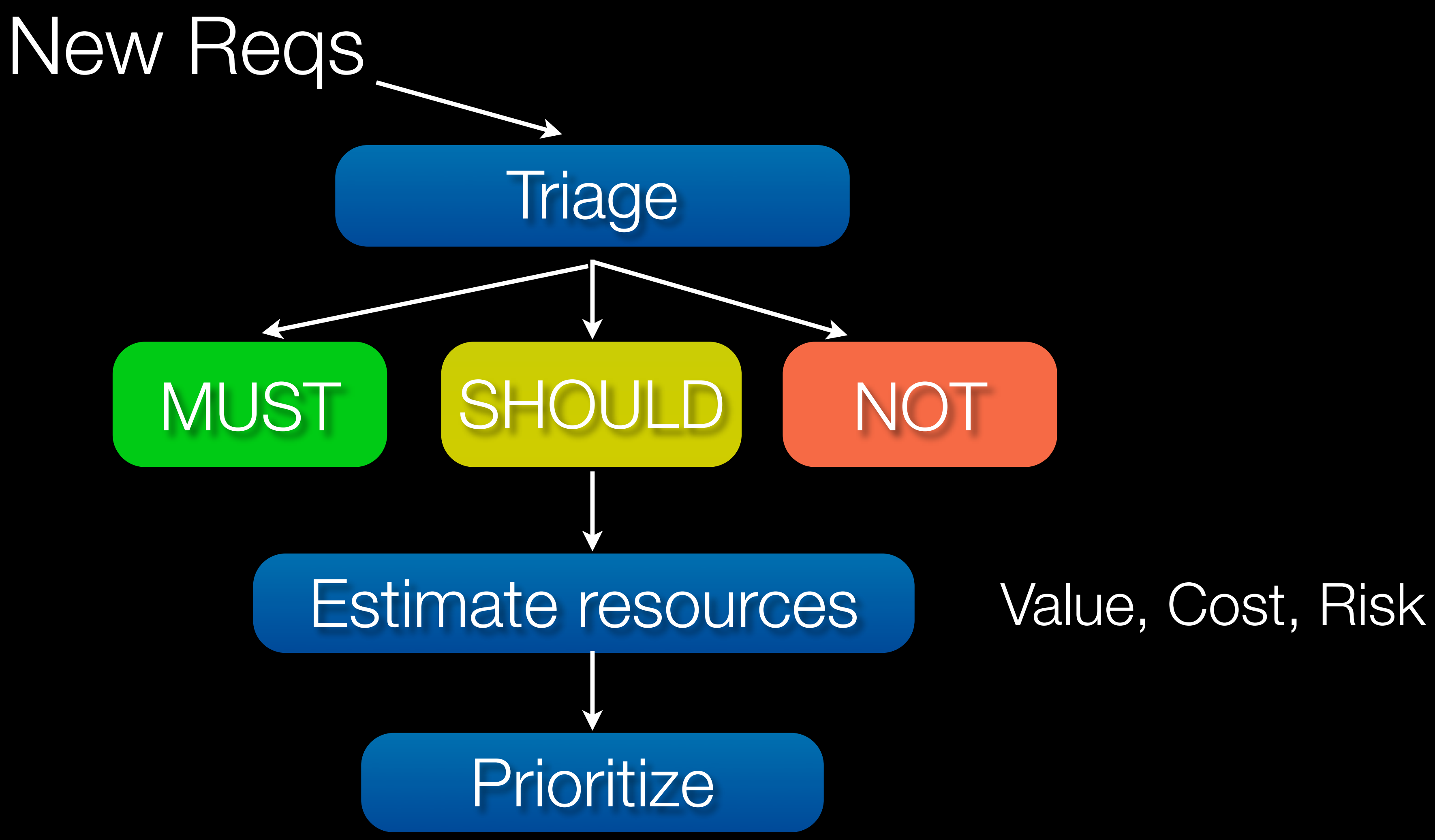


Triage (in MDRE)

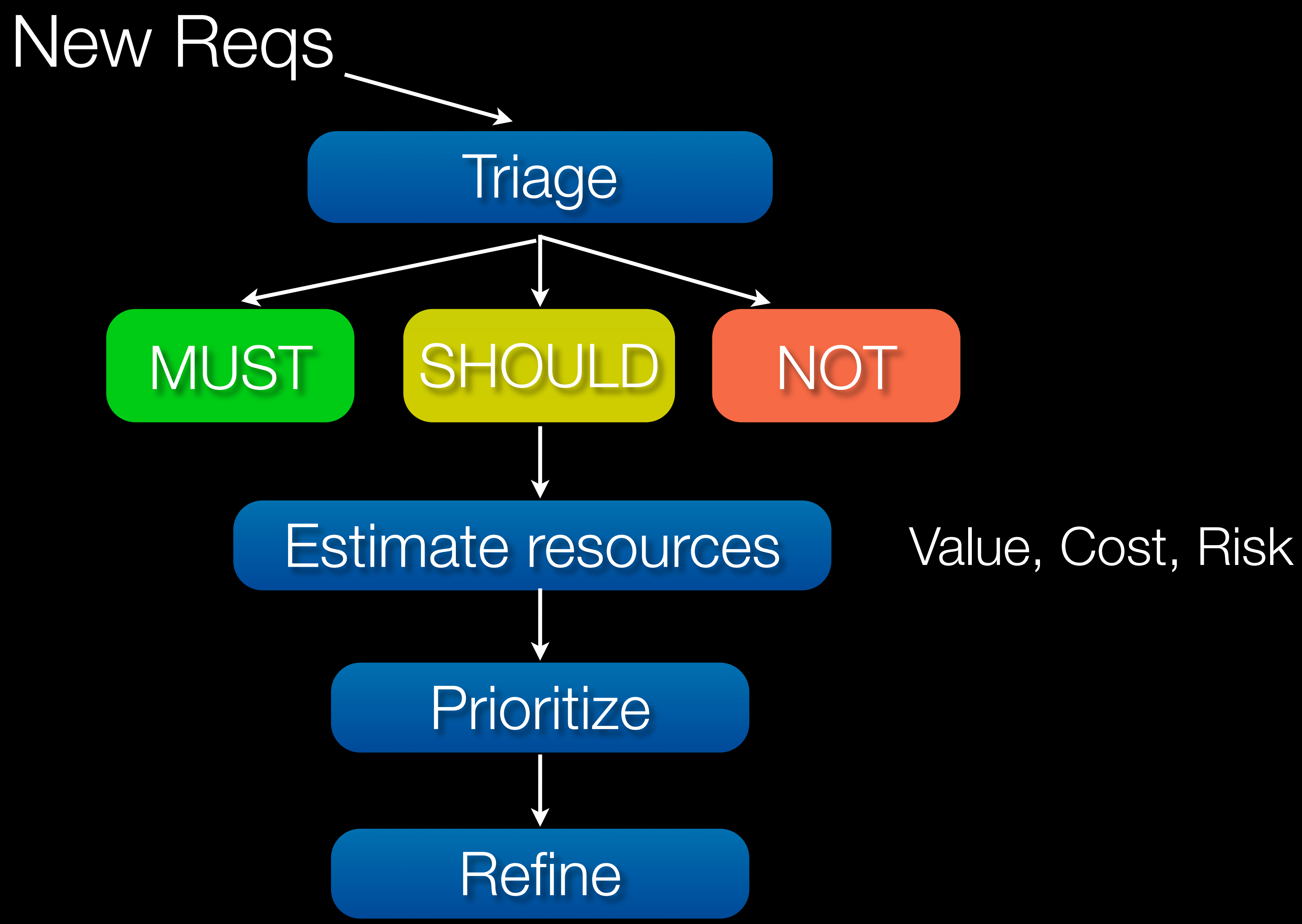


Value, Cost, Risk

Triage (in MDRE)



Triage (in MDRE)



New Reqs

Triage

MUST

SHOULD

NOT

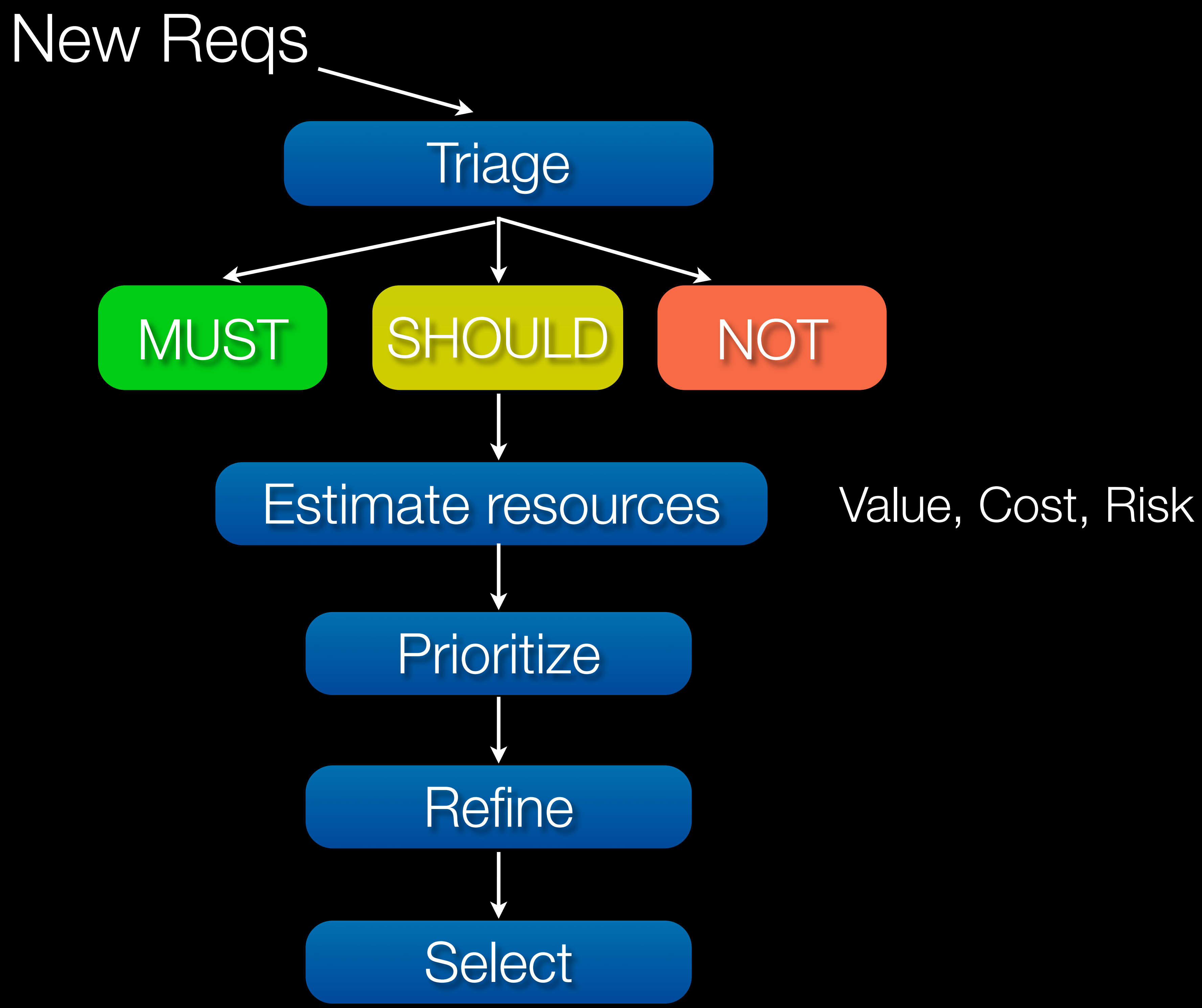
Estimate resources

Value, Cost, Risk

Prioritize

Refine

Triage (in MDRE)


New Reqs

Triage

MUST

SHOULD

NOT

Estimate resources

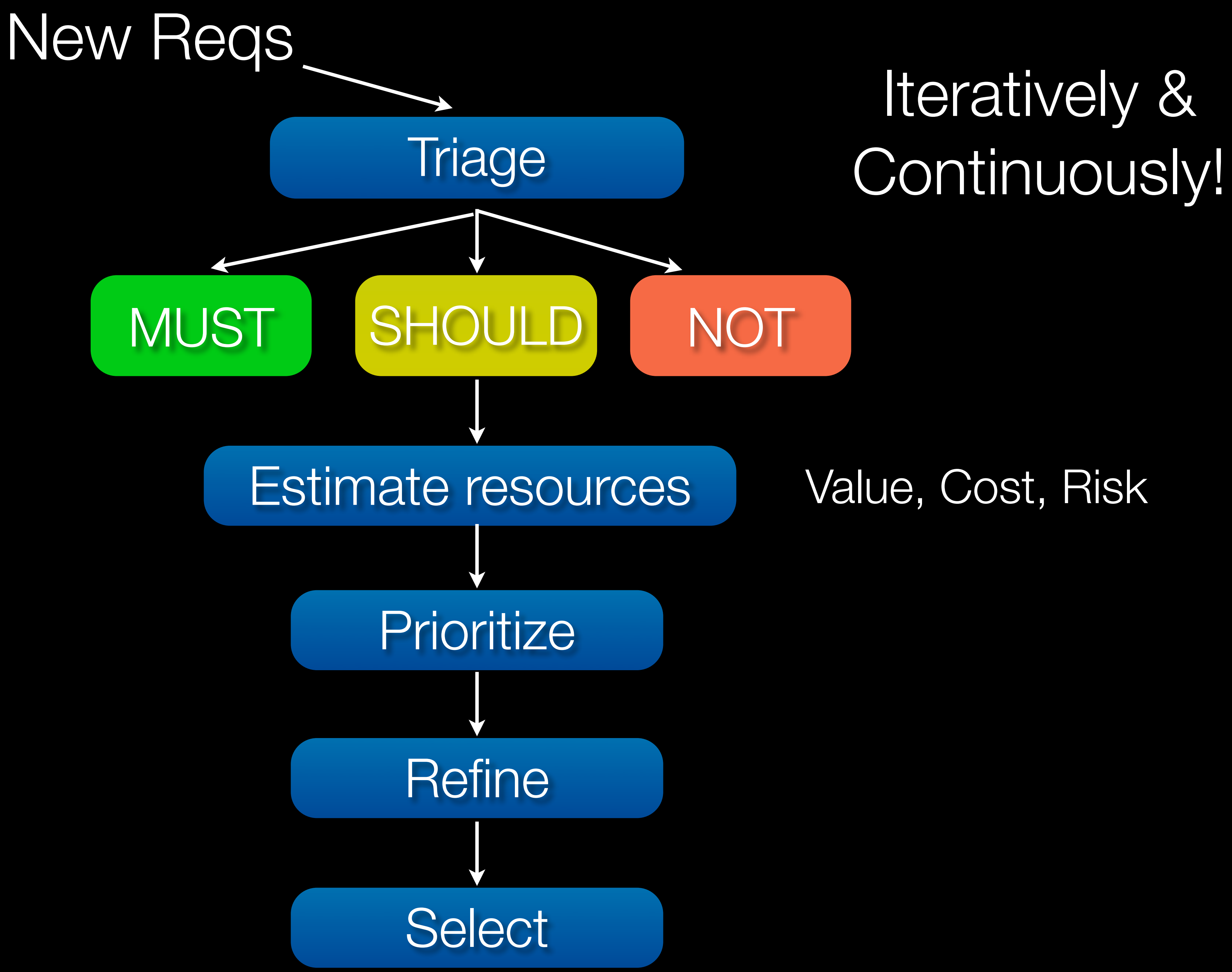
Value, Cost, Risk

Prioritize

Refine

Select

Triage (in MDRE)


New Reqs

Triage

MUST

SHOULD

NOT

Estimate resources

Value, Cost, Risk

Prioritize

Refine

Select

Iteratively & Continuously!

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

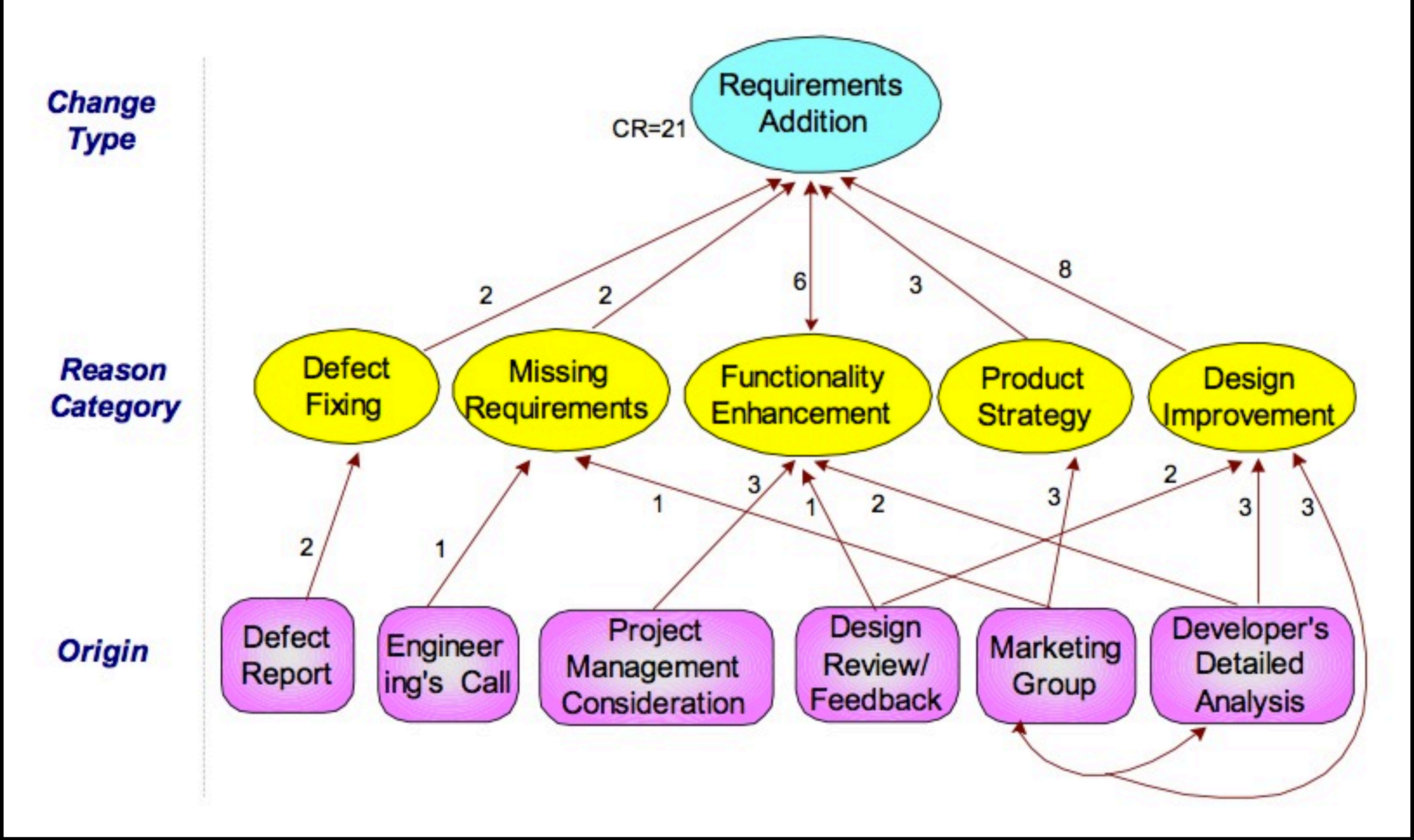
[Nurmuliani2004]

Taxonomy of Req Change

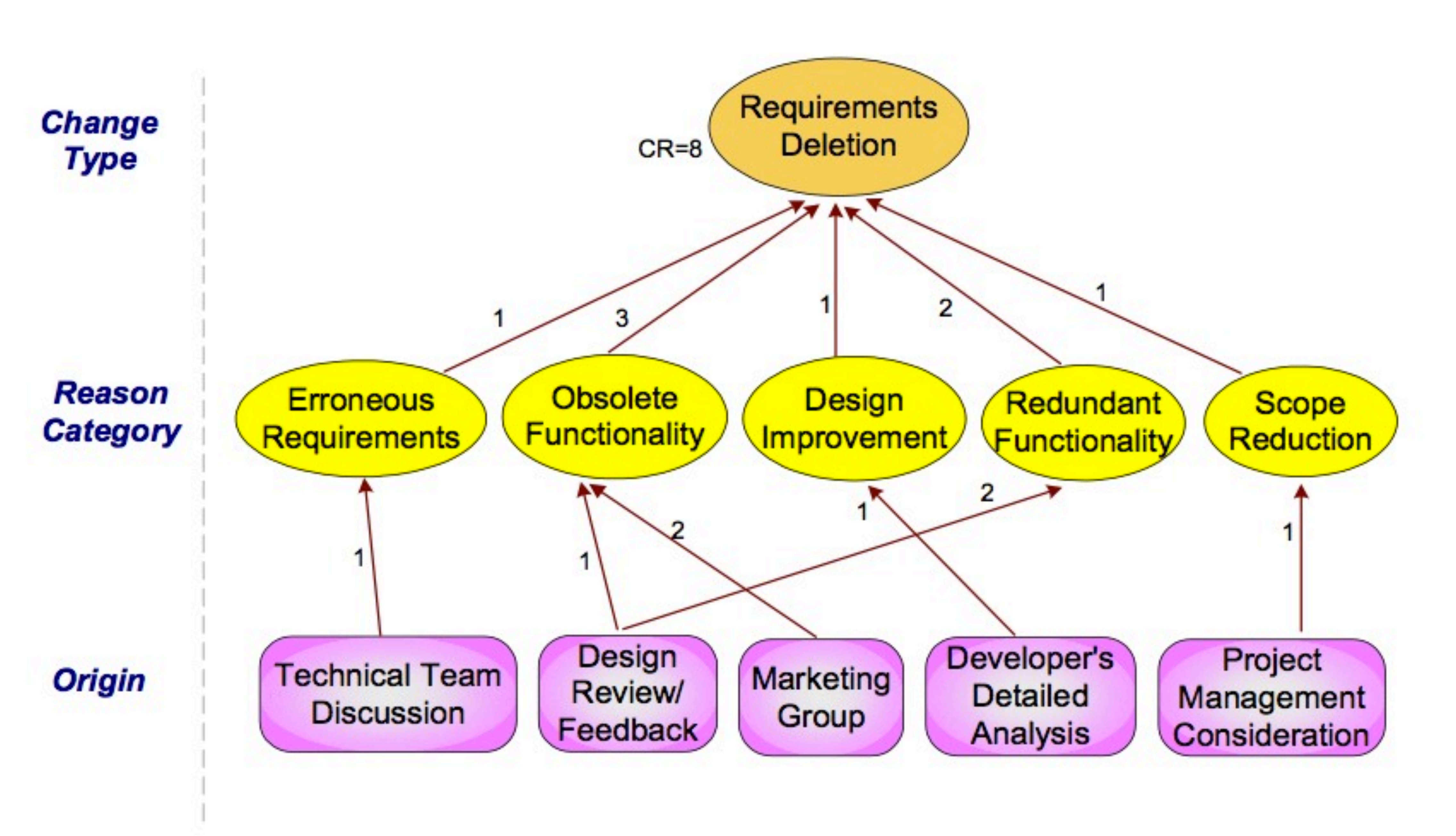
- Type: Add, Delete, Modify
- Reason: Defect fix, Func. Enhancement, Design improvement, ...
- Origin: Defect report, Marketing group, Review, ...

[Nurmuliani2004]

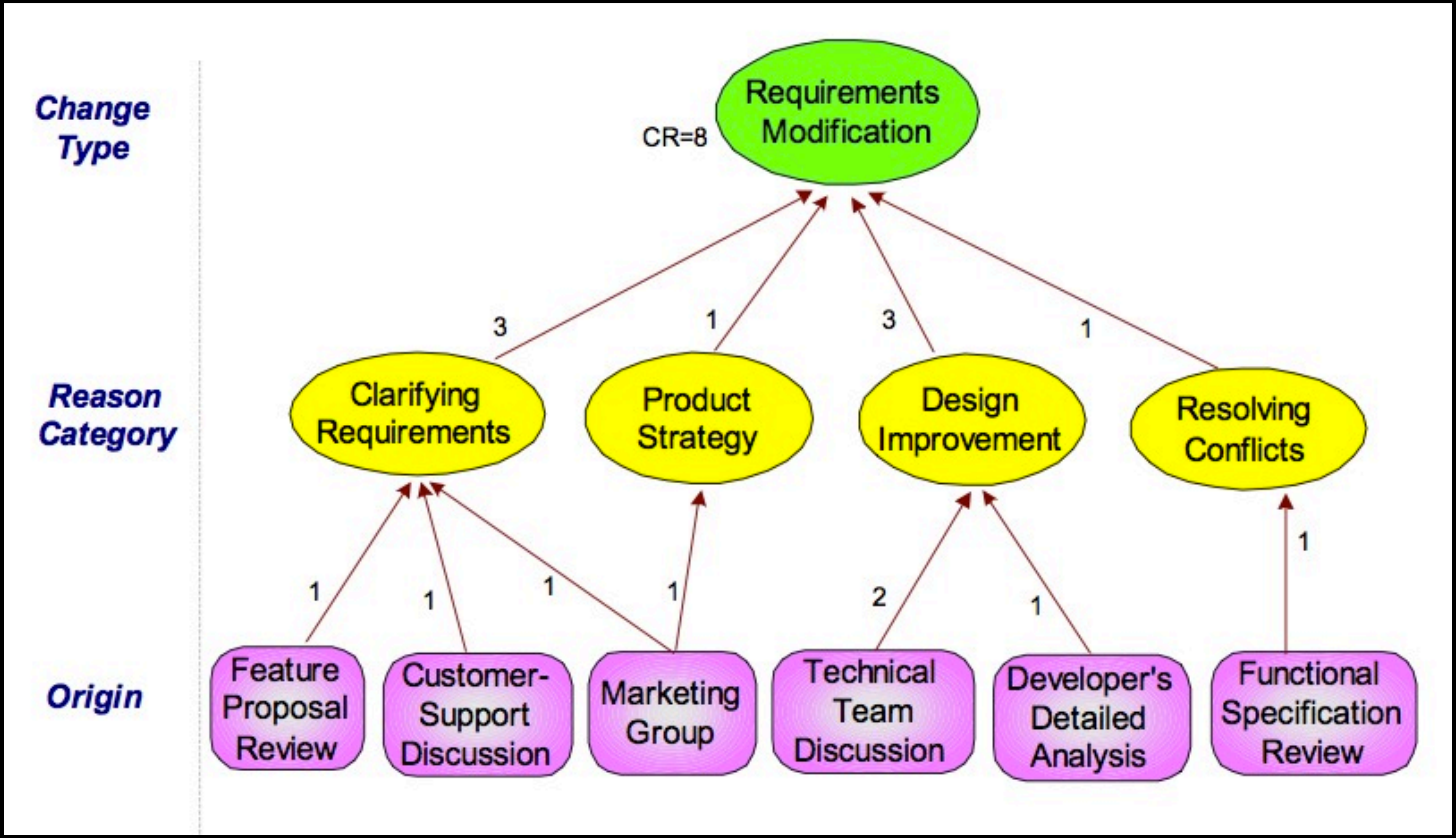
Requirement Additions



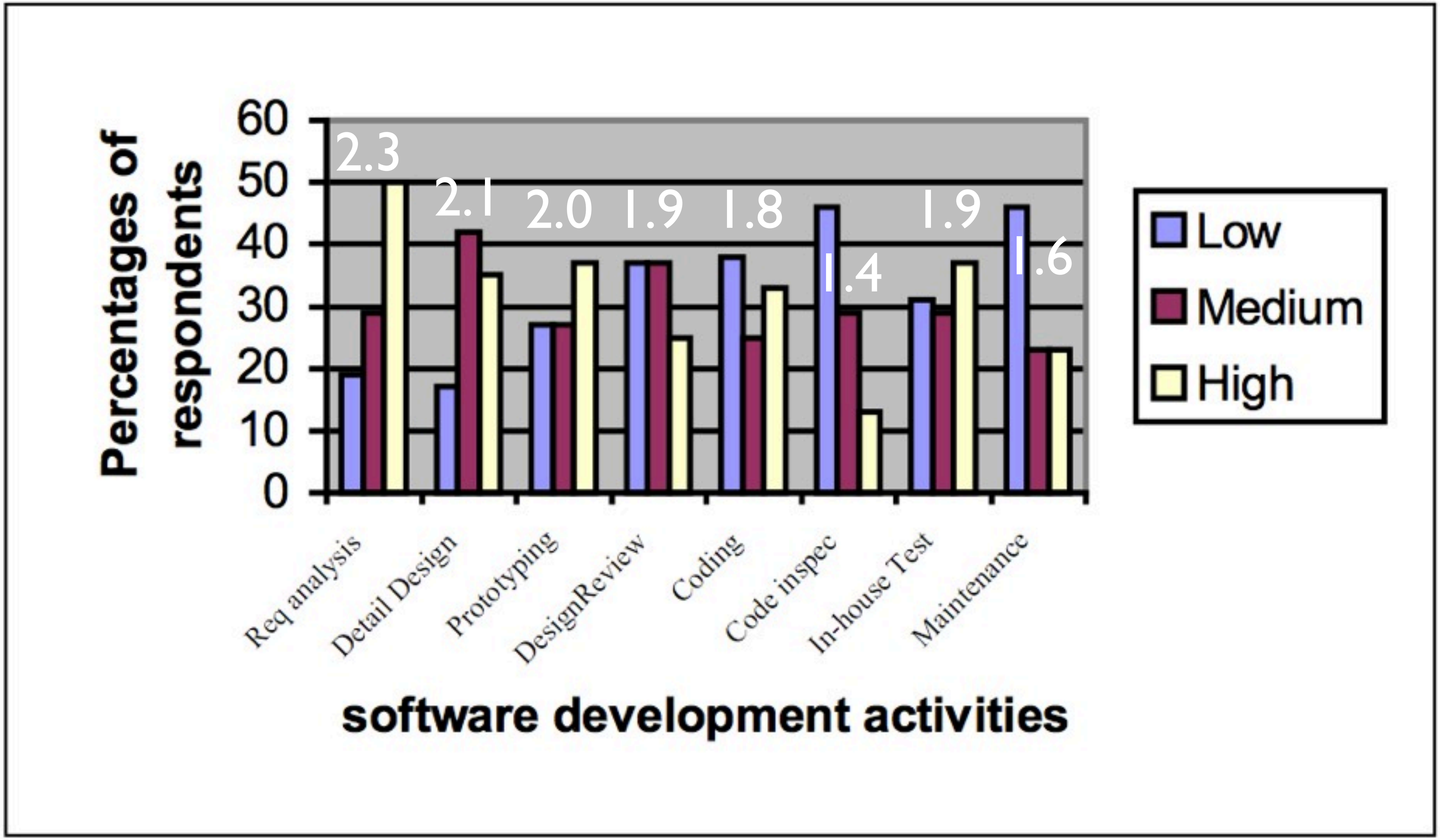
Requirement Deletions



Requirement Modifications



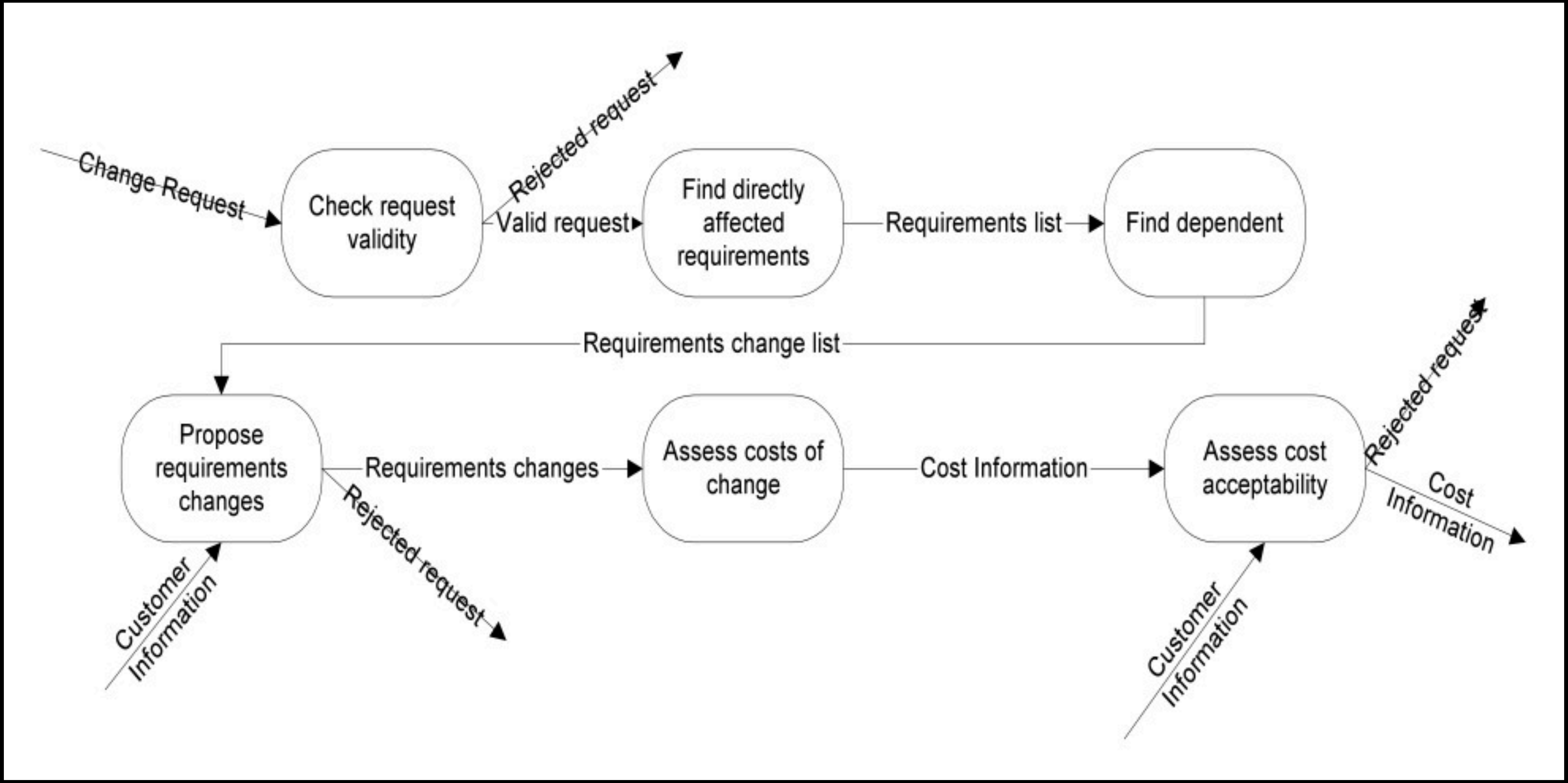
When are the changes?



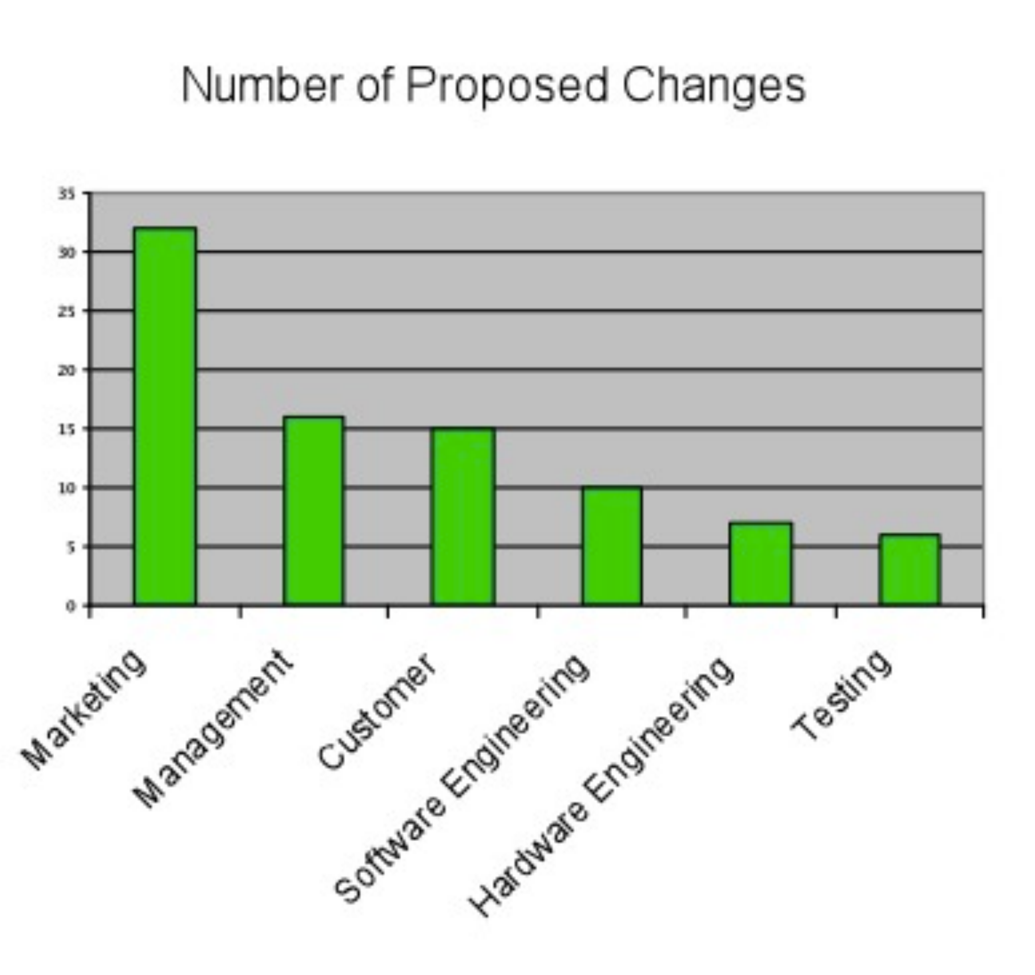
Preparing for change

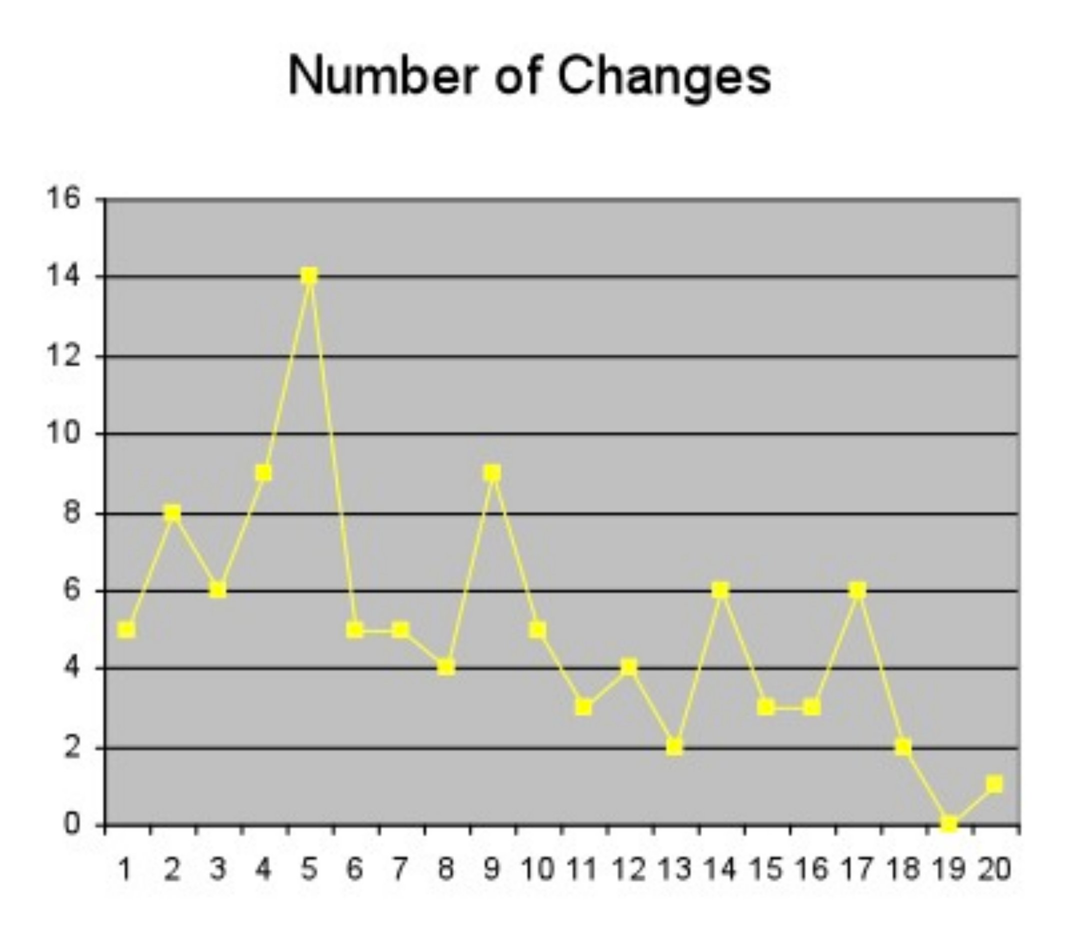
- Accept that changes are inevitable
- Identify Volatile Requirements
- Establish single channel for change requests
- Manage change hierarchically

Change Management Process



Measuring Change Activity



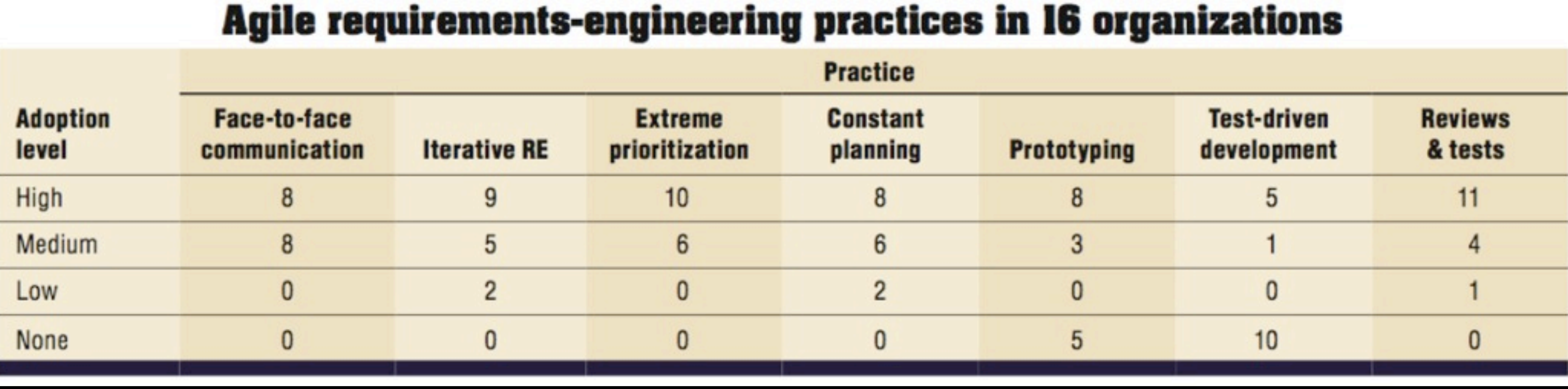


- Assess stability of requirements
- Identify improvement opportunities
- Alerts to project planning

“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	Requires trust
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

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

Iterative req engineering

- 🟩 Clearer reqs
- 🟩 Customer relation
- 🟥 Minimal docs when problems

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

Minimal docs when problems

Customer relation

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

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

Managing Change through Constant replanning

Pro	Con
Fewer changes	Architecture suffers
Smaller 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	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