

A Team,
A System,
Some Legacy
... and you

Eoin Woods
www.eoinwoods.info

Real Projects

- The books talk about building new systems
- Conferences are all about new technology
- How come you've got 2 million lines of Java 1.4 on WebLogic 8 with Oracle 9i?

*That's what most
of us have!*

leg·a·cy sys·tem | 'legəsē sistəm |

noun

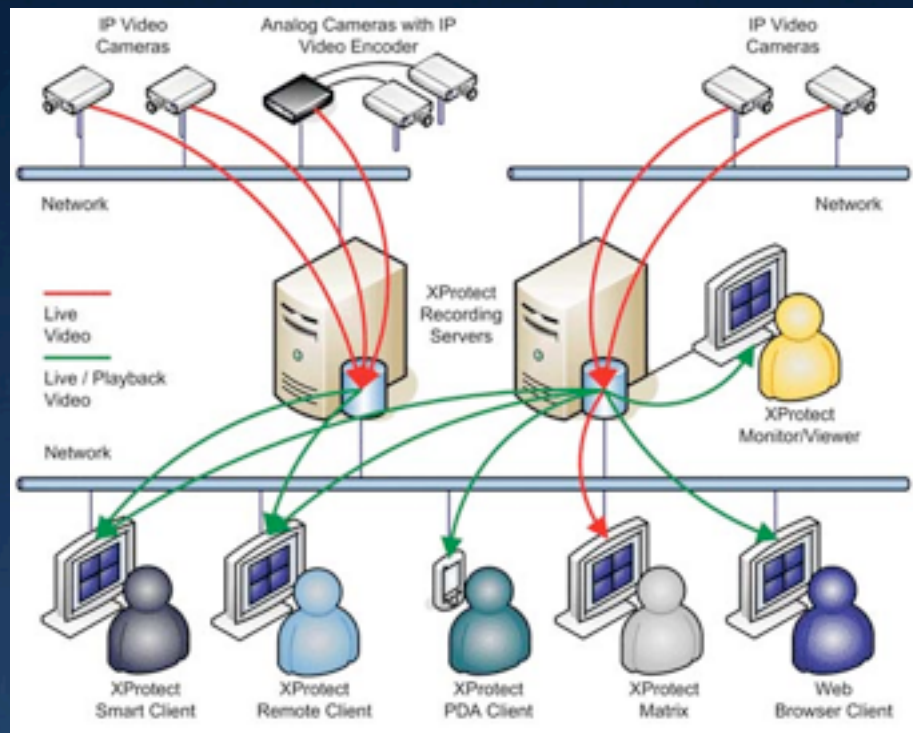
a system so valuable to the
organisation that nobody
dares to turn it off

Software Architecture with Real Teams

Being Late to the Party

- Software architecture often seems valuable only once things have gone wrong
- Architects often join existing projects to help improve difficult situations
- Always too much to do in the time available
- Often a real sense of urgency to “improve”

A Typical Situation



a system

and

a team



- All is not well with the system
- A new architect is told to “fix” “things”

What Could You Do?

Create models

Gap Analysis
of Functions

Replace
Difficult
Technology

Automated
Acceptance
Tests

HA/
Resilience
Improvements

Monitoring and
Alerting

Security
Assessment
Implement
AAA

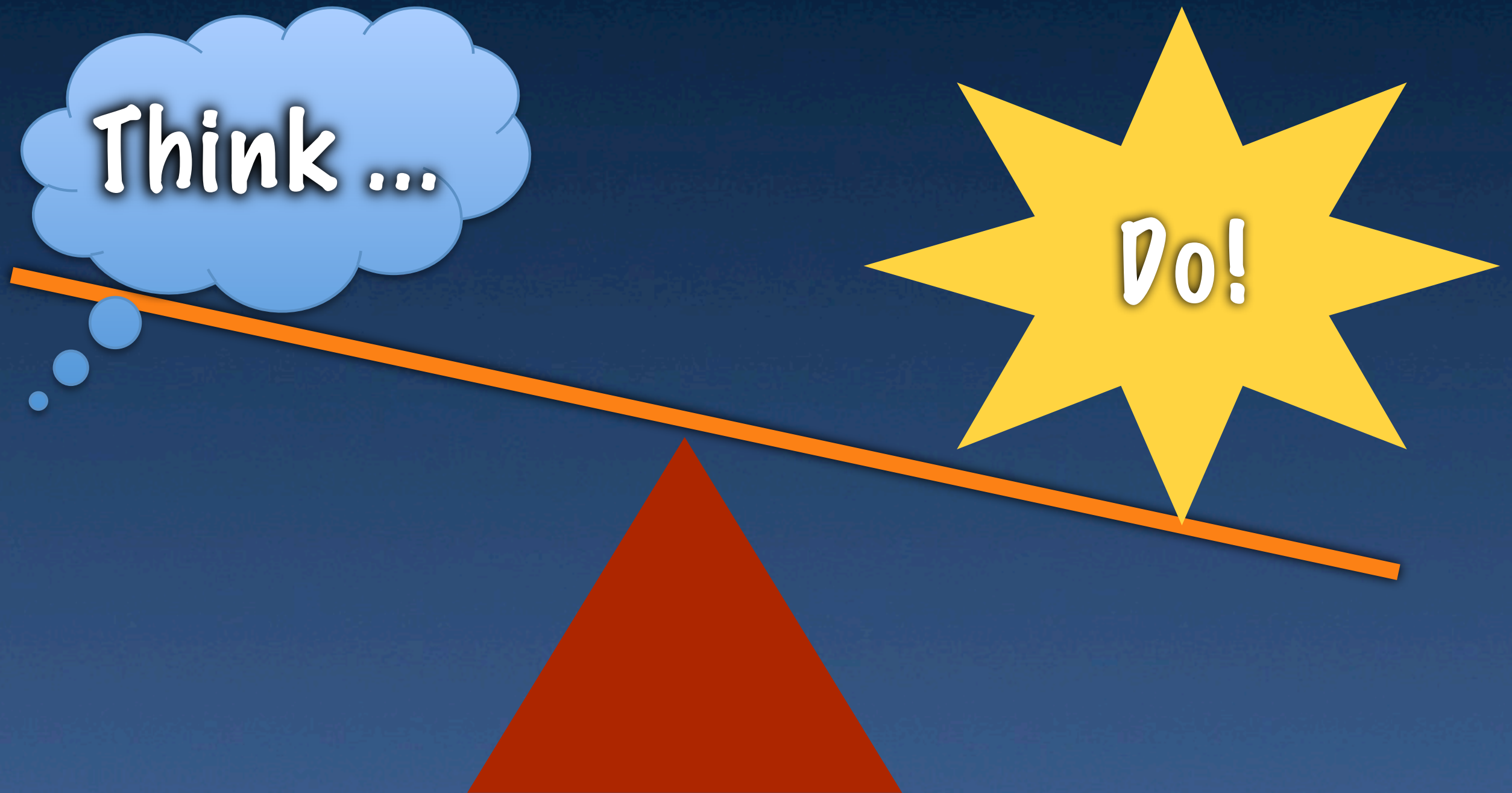
Meet 2 Year
Scalability
Goals

Continuous
Deployment

Refactor to
Patterns

All of this might make sense ... but you won't have time!

Inherent Tension



The Excesses to Avoid



Our Inspiration: The Master Builder

Who Were the Master Masons?

- The technical leads of their era
 - ... architect of the building, as administrative official of the building fabric, as building contractor, and finally, as technical supervisor of construction.
L.R. Shelby, The Role of the Master Mason, Speculum, Vol. XXXIX, 1964.
- Expert technologists, accomplished builders, proven leaders
- Vitruvius: “irmitas, utilitas, venustas”
 - sturdy, useful, beautiful



[Architects] who have aimed at acquiring **manual skill without scholarship** have never been able to reach a position of authority to correspond to their pains, while those who **relied only upon theories and scholarship** were obviously hunting the shadow, not the substance. But **those who have a thorough knowledge of both, like men armed at all points, have the sooner attained their object and carried authority with them.**

Marcus Vitruvius Pollio
De Architectura (“The Ten Books on Architecture”)



Master Masons in the Mud

- Easy to build an ivory tower in a green field
- Brown field projects need immediate help
- Long term thinking is good
 - but things need attention RIGHT NOW
- Need the broad skills of the master mason
 - not afraid of theory or practice

Architects on Brown-Field Projects

Finding Your Bearings

*Minimal
Modelling*

*Consider the
Team*

*Assessment
Techniques*

*Monitor and
Measure*

*Automated
Analysis
Tools*

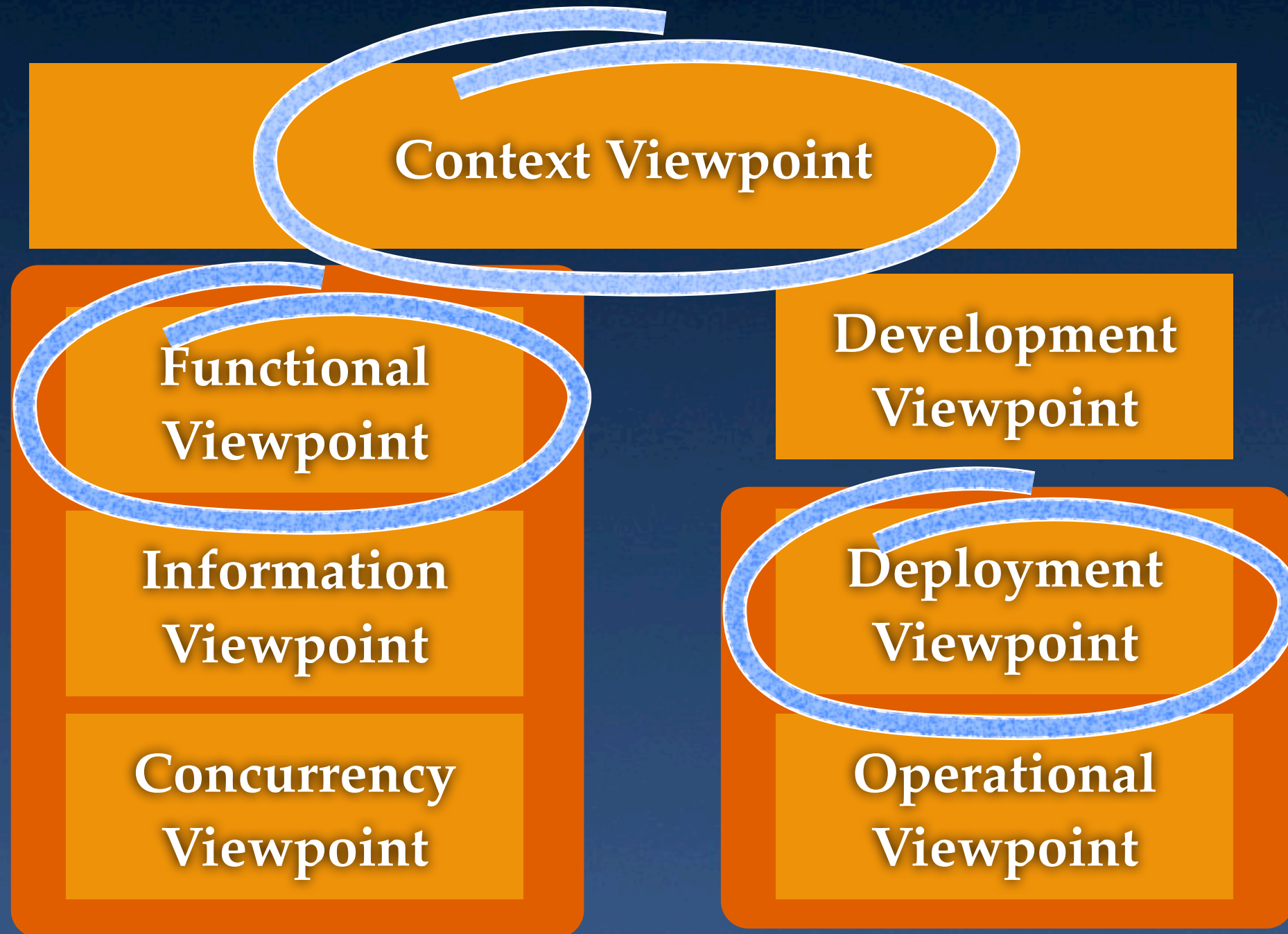


*"If you cannot measure it, you cannot improve it."
Lord Kelvin*

Getting the Right Perspective

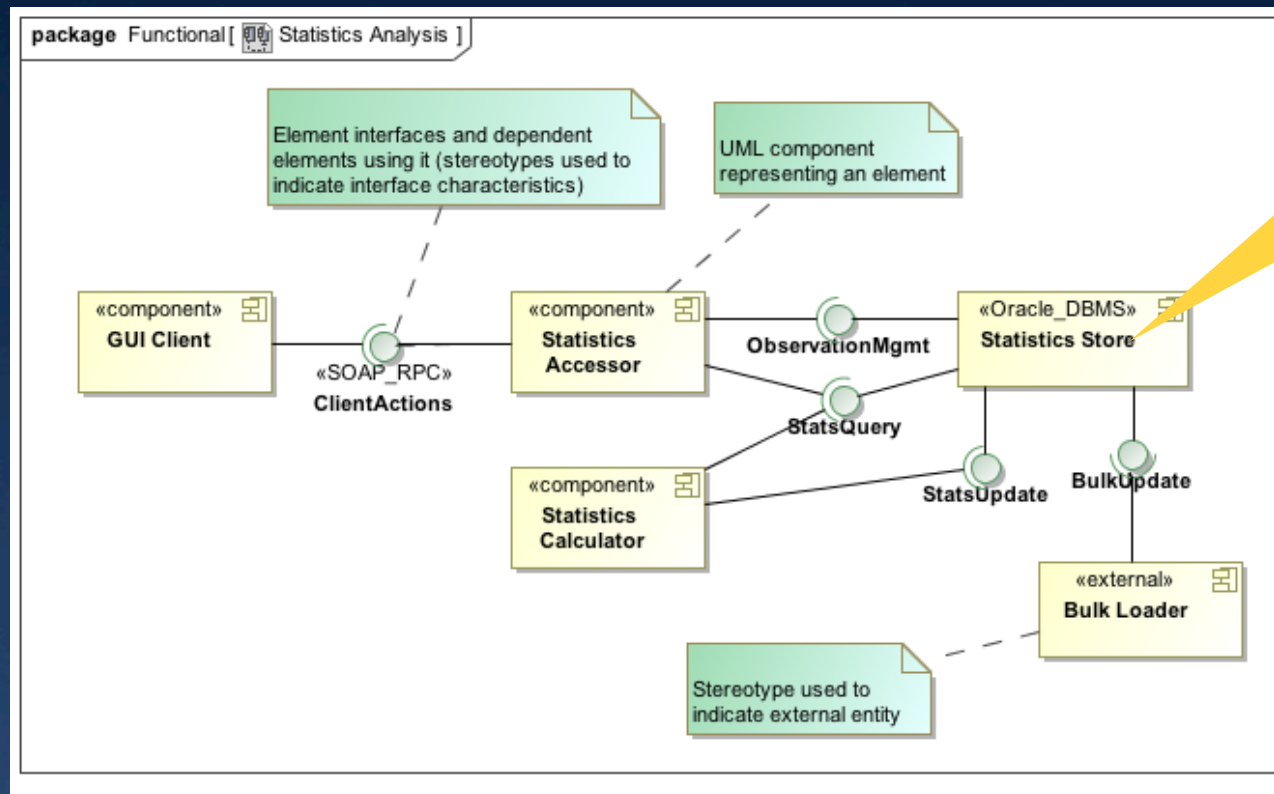


Minimal Modelling



Capture what you can't get from the code

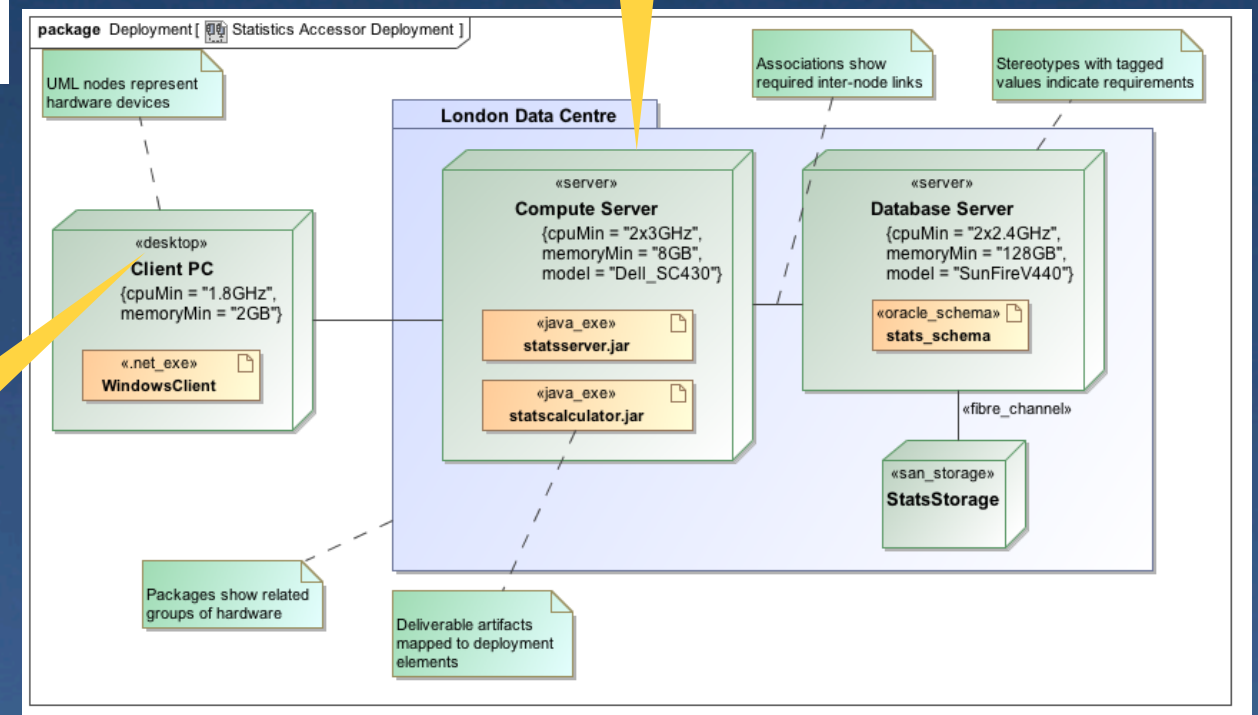
Minimal Modelling



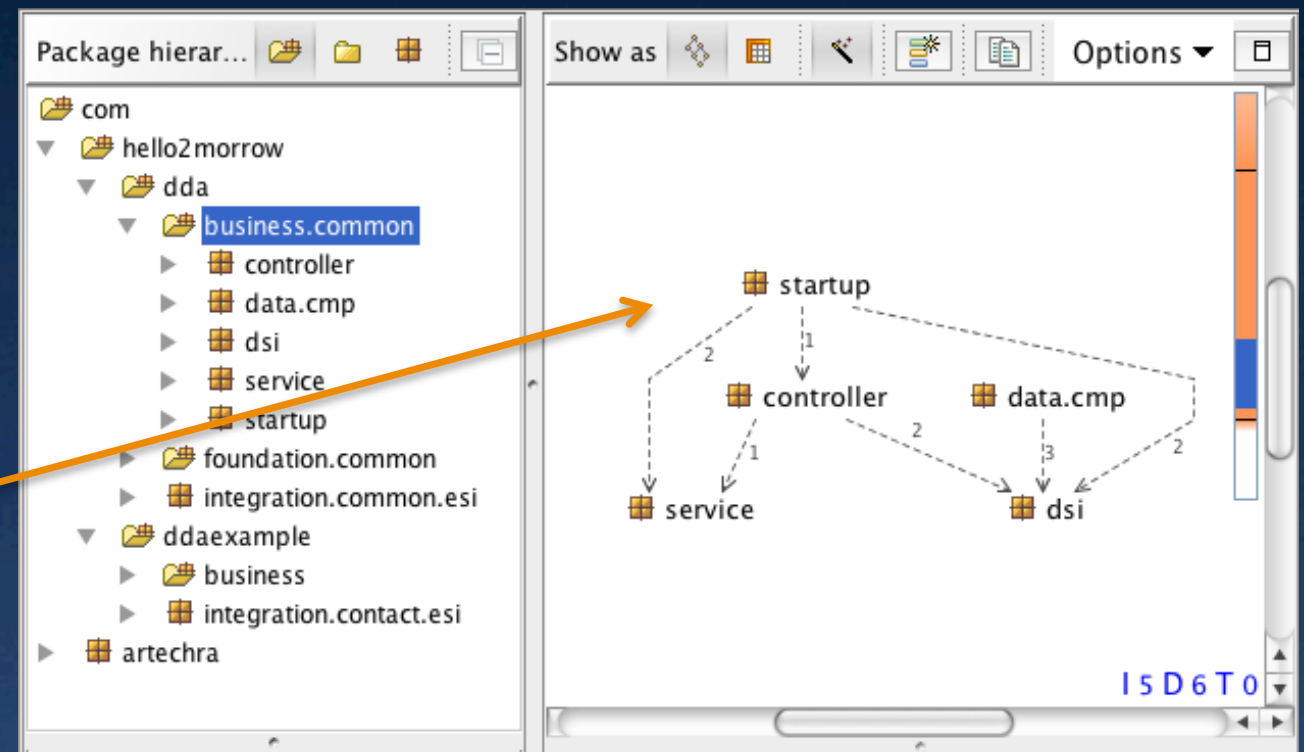
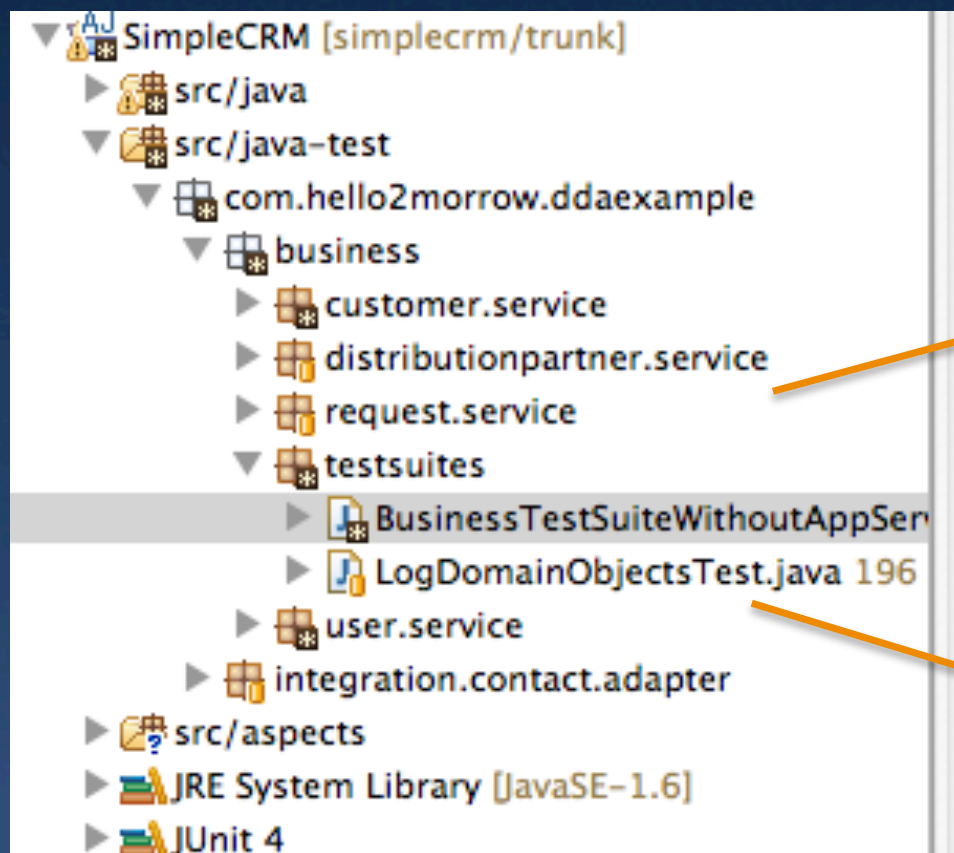
Small well
annotated models

Define your
notation!

Focus on essentials that
someone needs to know



Automated Analysis



Package hierar...

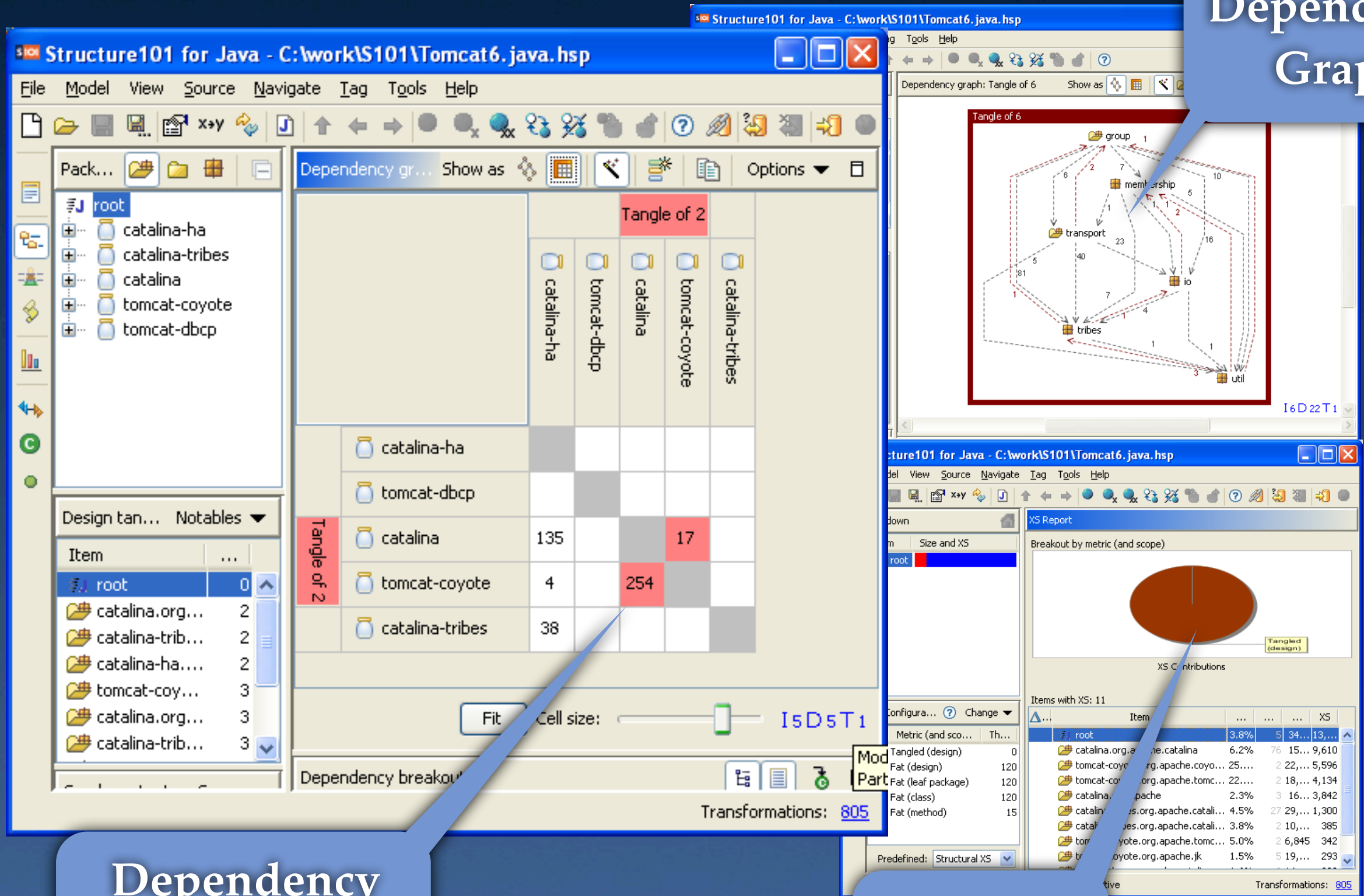
- com
 - hello2morrow
 - dda
 - business.common
 - controller
 - data.cmp
 - dsi
 - service
 - startup
 - foundation.common
 - integration.common.esi
 - EsiFactory
 - ExternalSystemIf
 - ddaexample
 - business
 - contact
 - customer
 - distributionpartner

Show as Options

	common.startup	distributionpartner	request	customer	contact	user
common.startup						
distributionpartner						
request		33				
customer		11				
contact		18		20		
user		4	4	4		

Fit Cell size: 16 D 7 T 0

Tooling: S101, Lattix, Sonar, ...



Dependency Graph

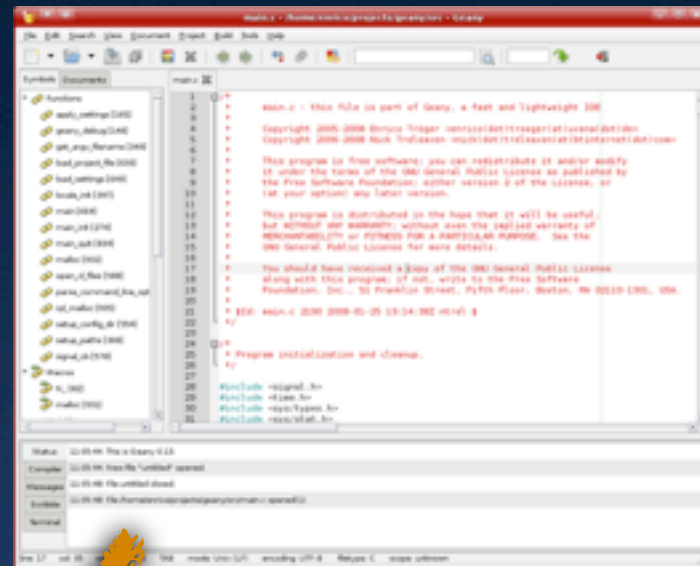
Dependency Matrix

Metrics

Monitor and Measure



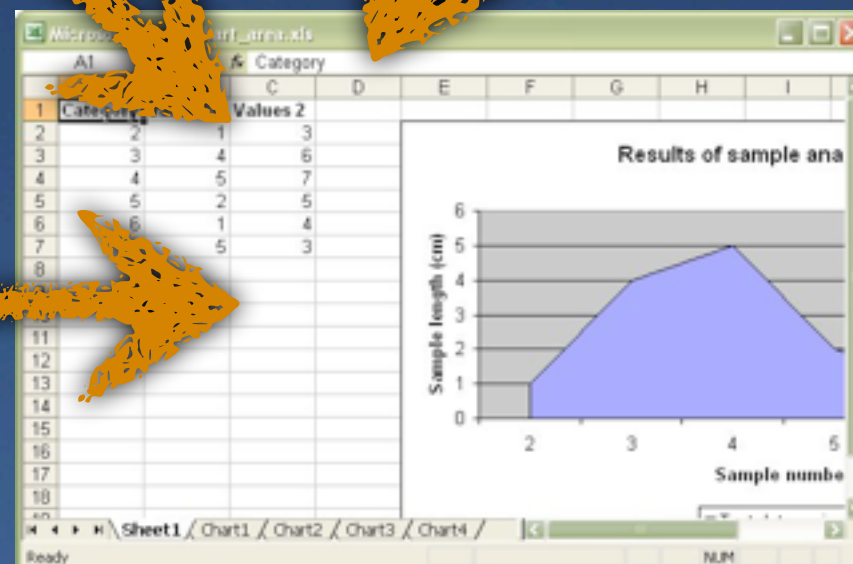
Production
Metrics



Implementation
Metrics

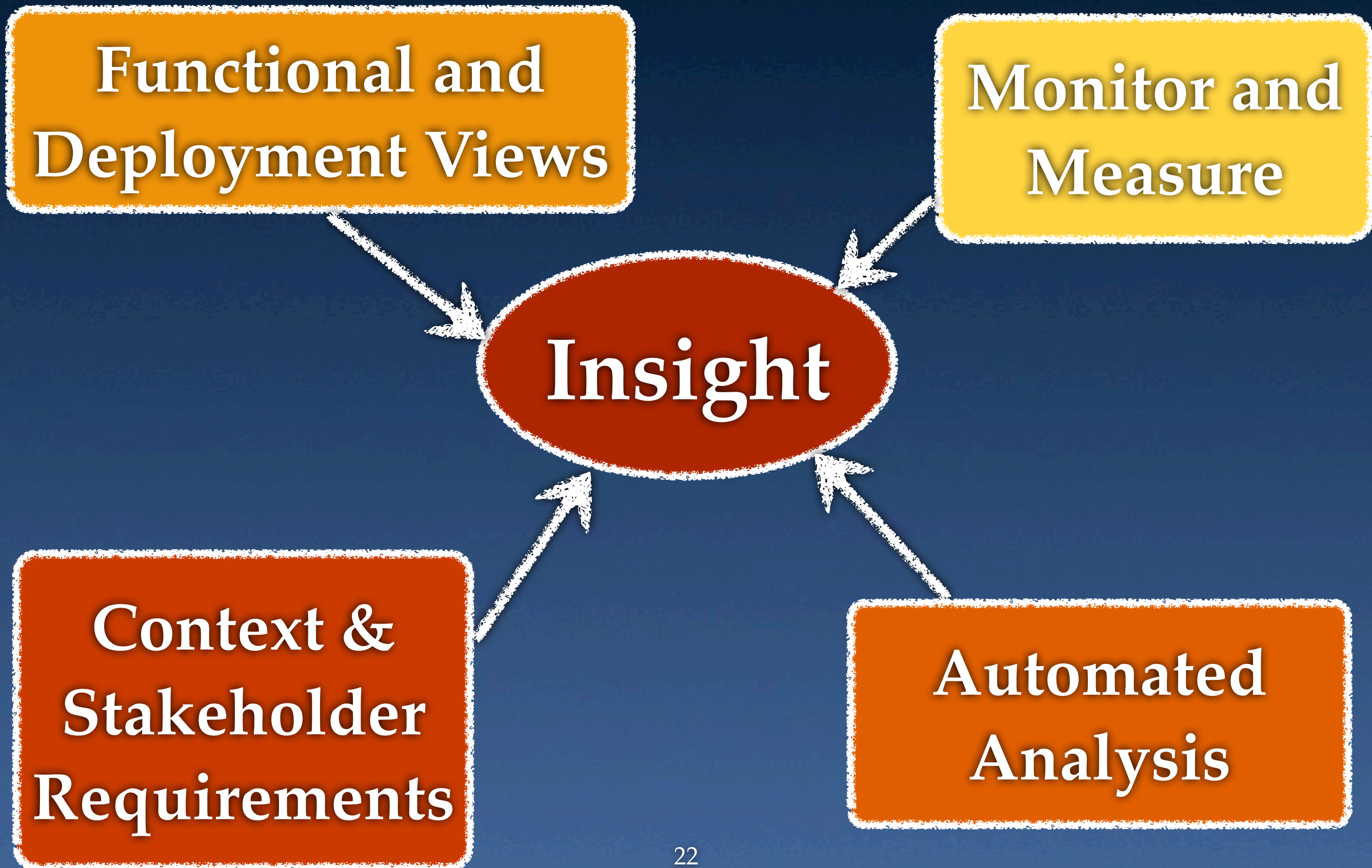


Stakeholder
Opinions



System
Qualities
Assessment

Assessment



Architectural Assessment - Pointers

- ATAM

- Architectural Tradeoff Analysis Method
- SEI method - search “ATAM”

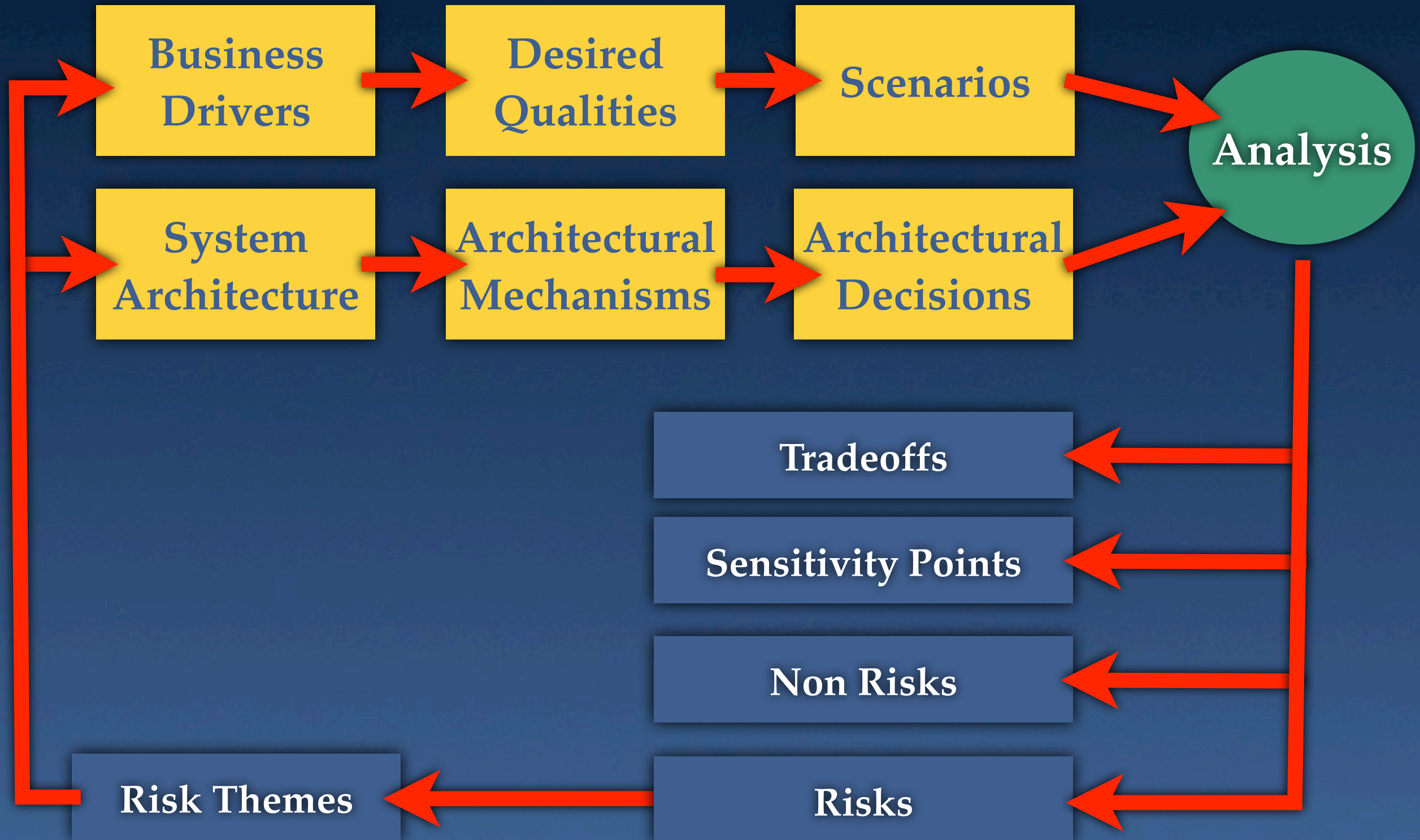
- LAAAM

- Lightweight Architectural Assessment Method
- Jeromy Carriere - search “LAAAM”

- TARA

- Tiny Architectural Review Approach
- Eoin Woods - <http://tiny.cc/tara-approach>

ATAM - Scenario Based Assessment

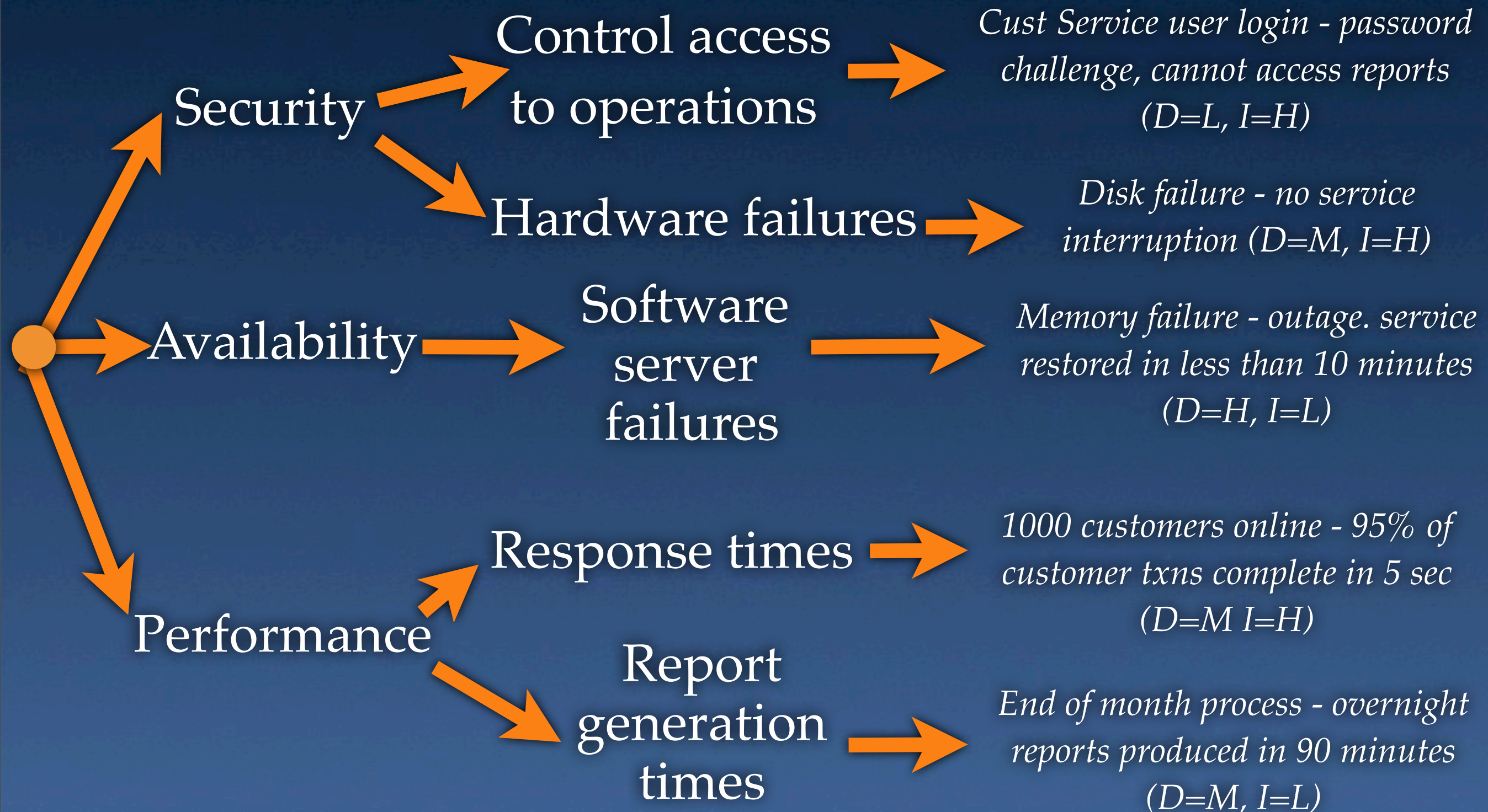


Quality Attributes and Scenarios

Qualities

Specifics

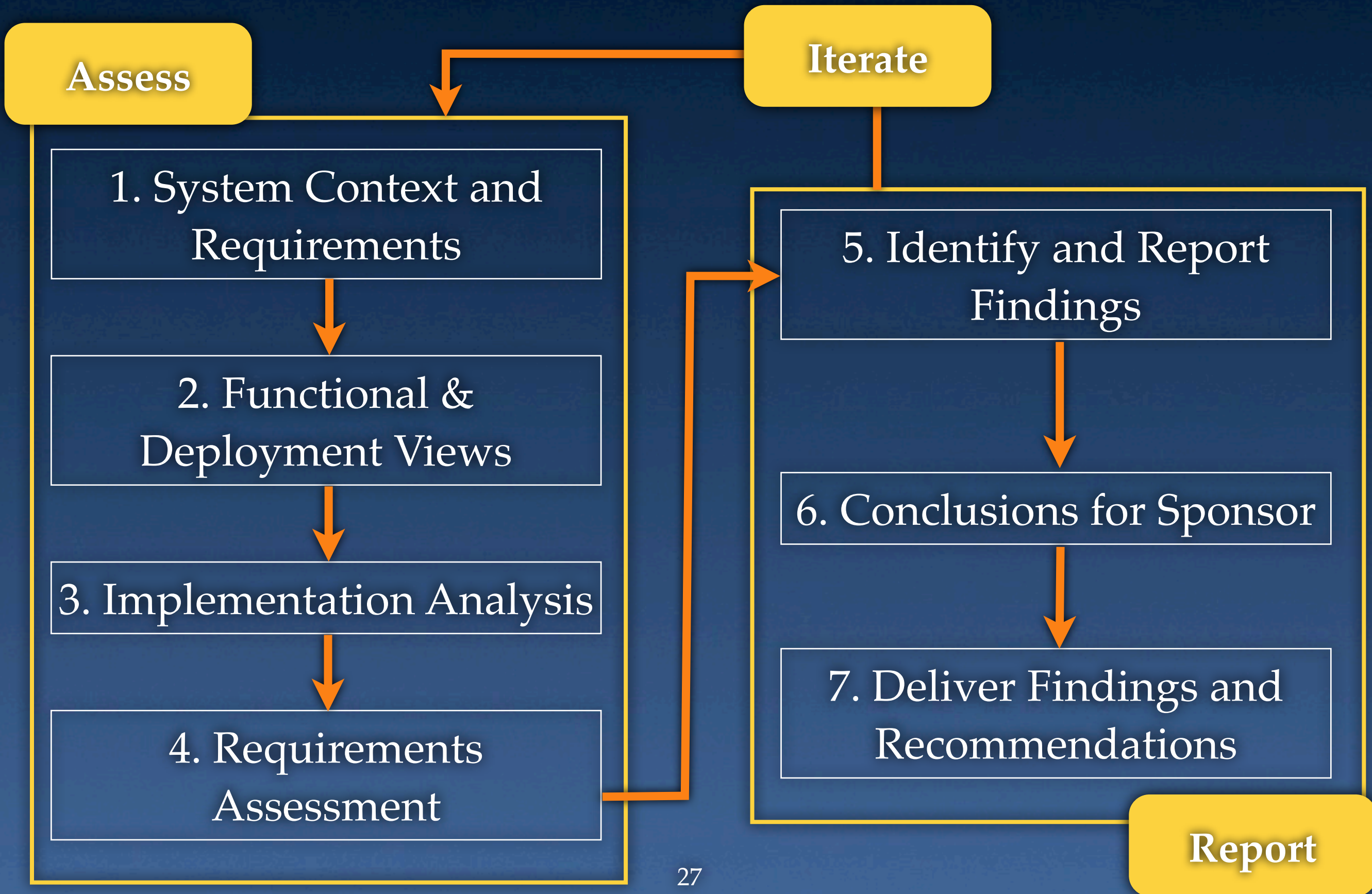
Scenarios



Example Scenario

- Context:
 - 1000 users actively using the system
 - No batch processing active
- Stimulus:
 - Search for customer by postcode initiated
- Response:
 - Initial pick list returned in 5 seconds
 - List navigation operations return in 2 seconds

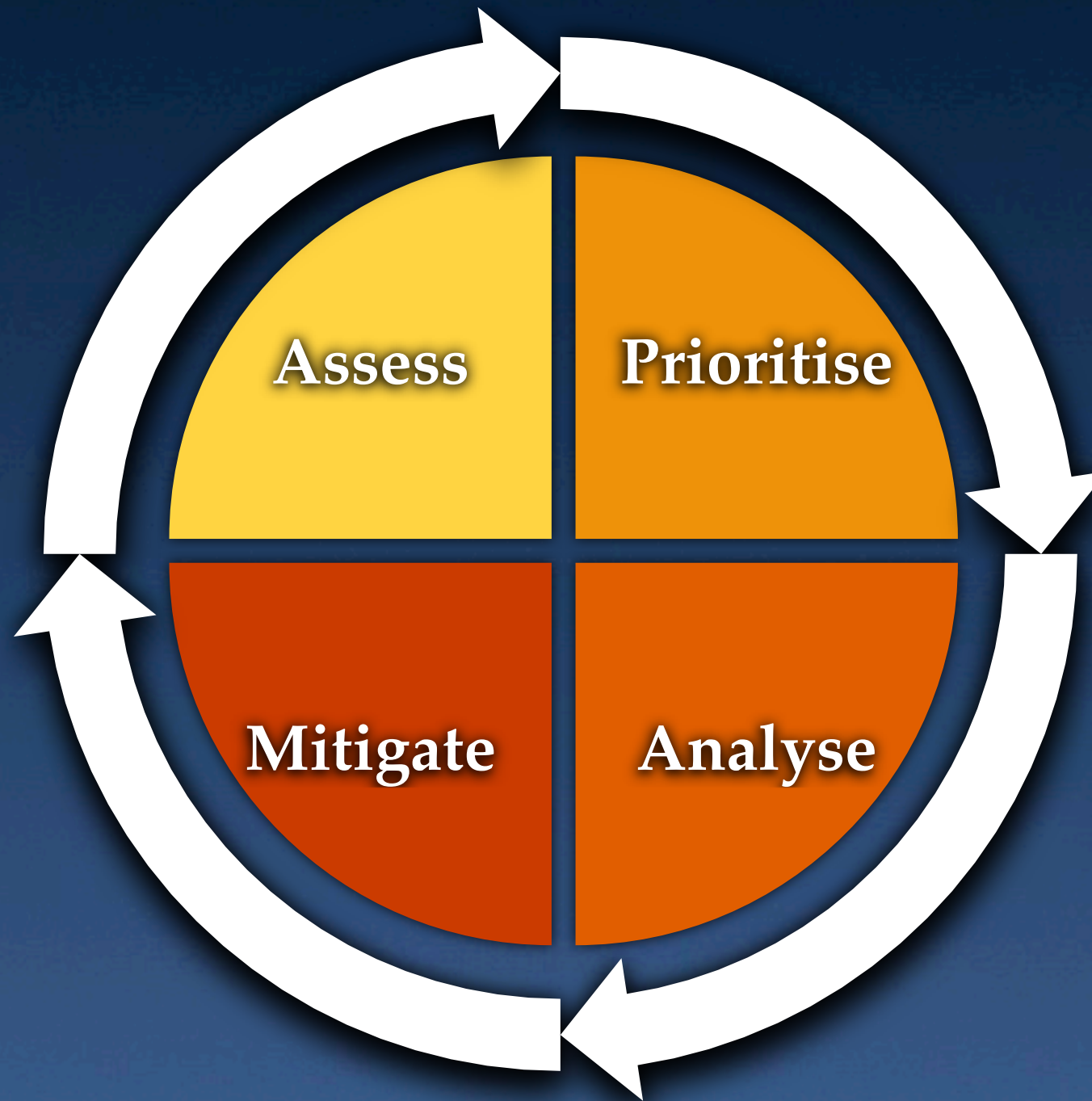
TARA - Simpler System Assessment



Consider the Team

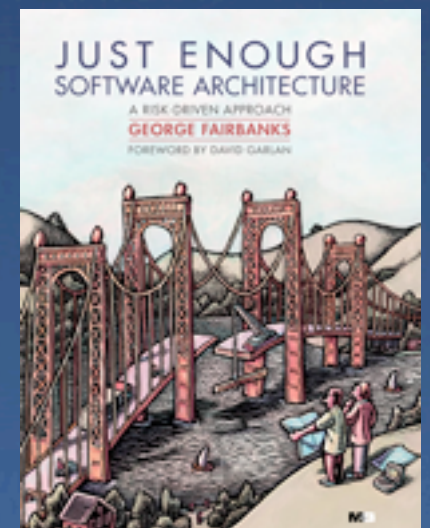
- Easy to focus on the technology and system
- The team probably need attention too
 - Morale?
 - Dynamics?
 - Confidence?
 - Competence?
- The team must shape your whole approach
 - otherwise risk goes sky high

Making Choices Based On Risks



How?

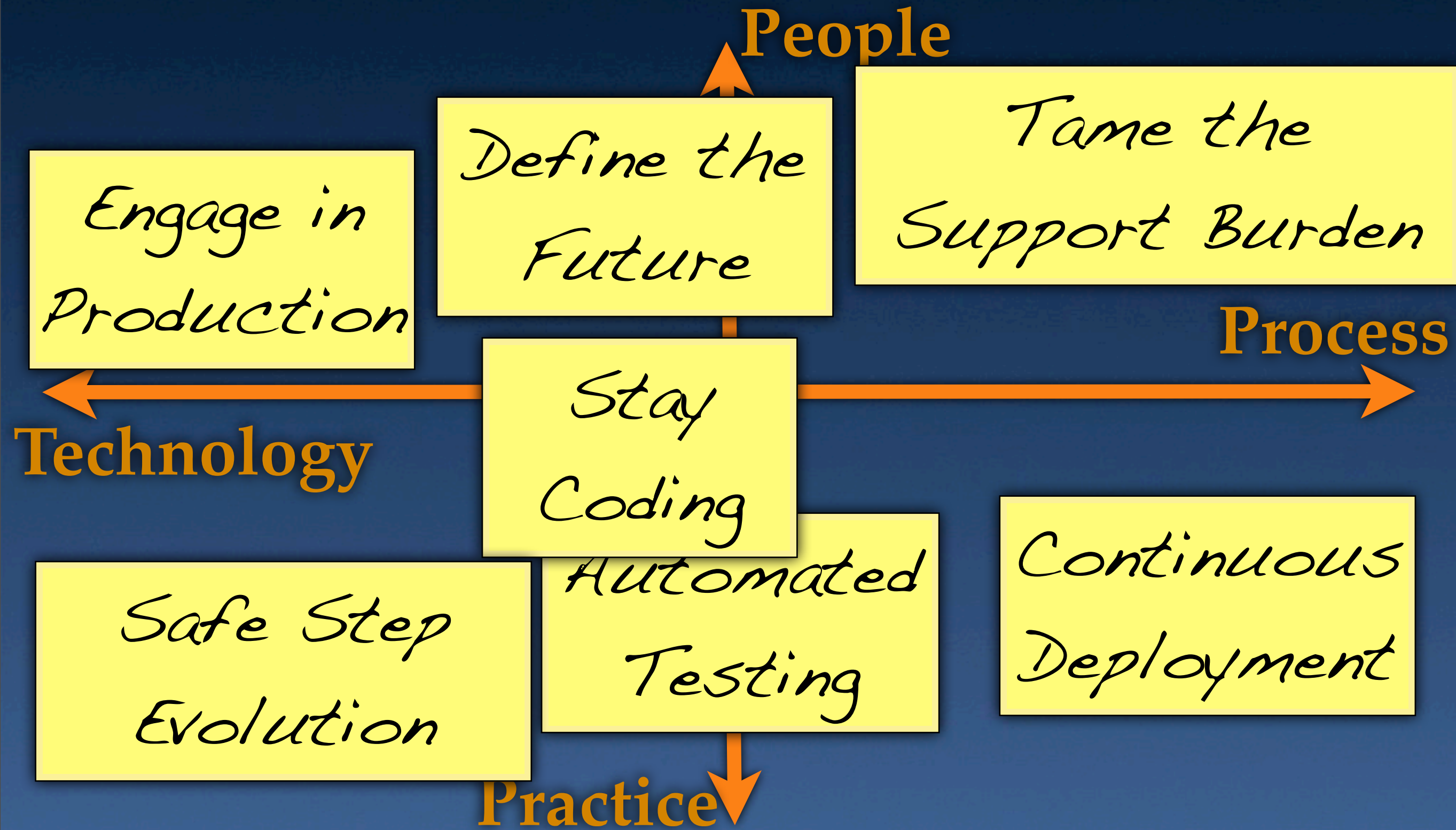
“Just Enough Software Architecture”
George Fairbanks



Tactics for Existing Projects

... or “WHAT WOULD VITRUVIUS HAVE DONE?”

Tactics for Existing Projects



Engage in Production



- Why?
 - reality check - rich information source
- How?
 - monitoring + stats + incident management
- For Who?
 - support, end-users, business management
- Pitfalls?
 - this is not your main job!

Tame the Support Burden



- Why?
 - support will sap the team of energy
- How?
 - stability first, then “BAU” effort (L2 team?)
- For Who?
 - end users, dev team, IT management
- Pitfalls?
 - but avoid “over the wall” mentality

Continuous Integration and Deploy

- Why?
 - efficiency and reliability
- How?
- For Who?
 - development & support teams
- Pitfalls?
 - running before you can walk, underestimation



Automated Testing



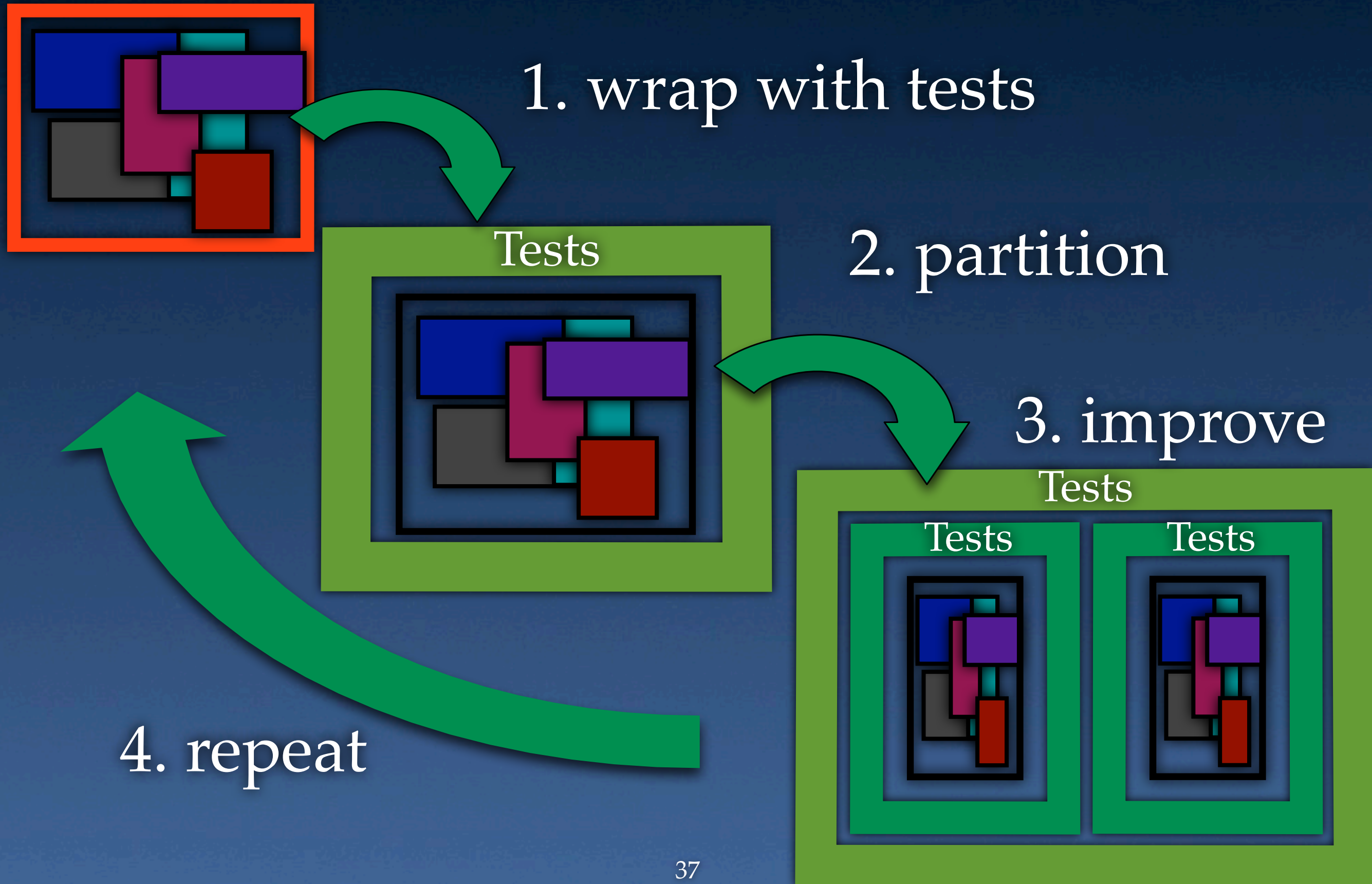
- Why?
 - confidence, efficiency + reveal problems
- How?
 - unit test + coverage, regression tests
- For Who?
 - everyone!
- Pitfalls?
 - tar pit of legacy (cost)

Safe Step Evolution

- Why?
 - control risk while improving
- How?
 - wrap with tests, partition, improve, ... repeat
- For Who?
 - everyone
- Pitfalls?
 - assumptions, knowledge gaps



Safe Step Evolution



Safe Step Evolution Tactics

partition

replace

simplify

extract

encapsulate

generalise

Stay Coding

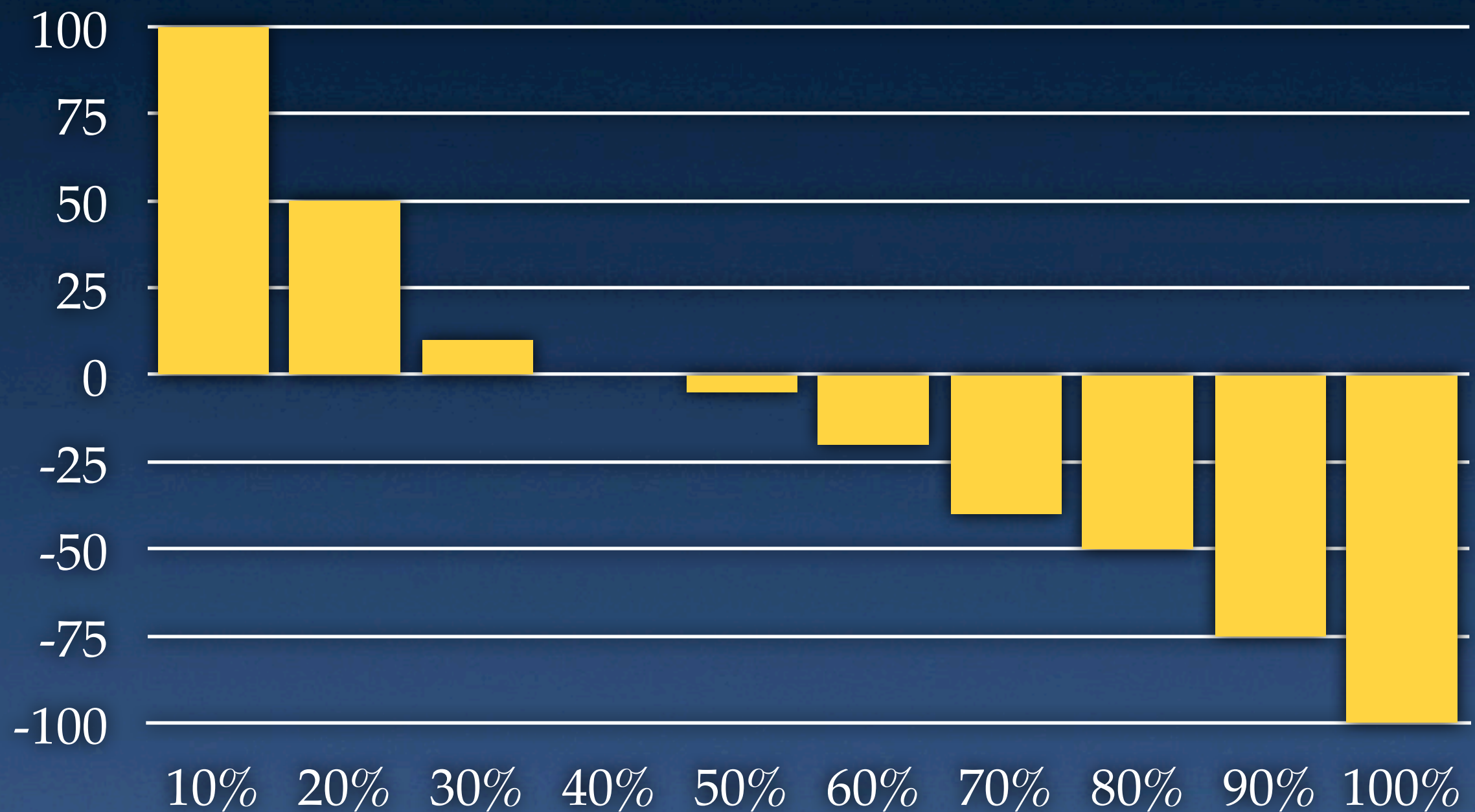


- Why?
 - see dev reality, stay current and credible
- How?
 - fix bugs, write tests, refactor, ... off critical path
- For Who?
 - you mainly!
- Pitfalls?
 - huge time sink - potentially low ROI

Code, Model ... Model, Code

- Coding time always gets squeezed (rightly)
- Code to build credibility
- Code to set an example
- Code when you're genuinely best for a task
- Code occasionally for sanity!

Notional ROI for Architect Coding



Where is your break-even point?

Define the Future



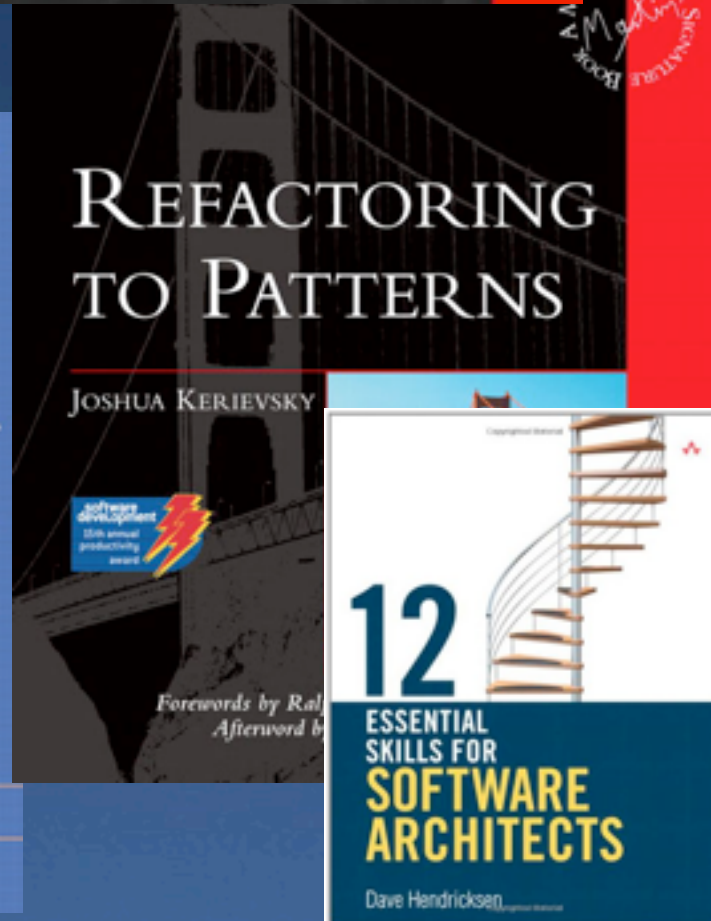
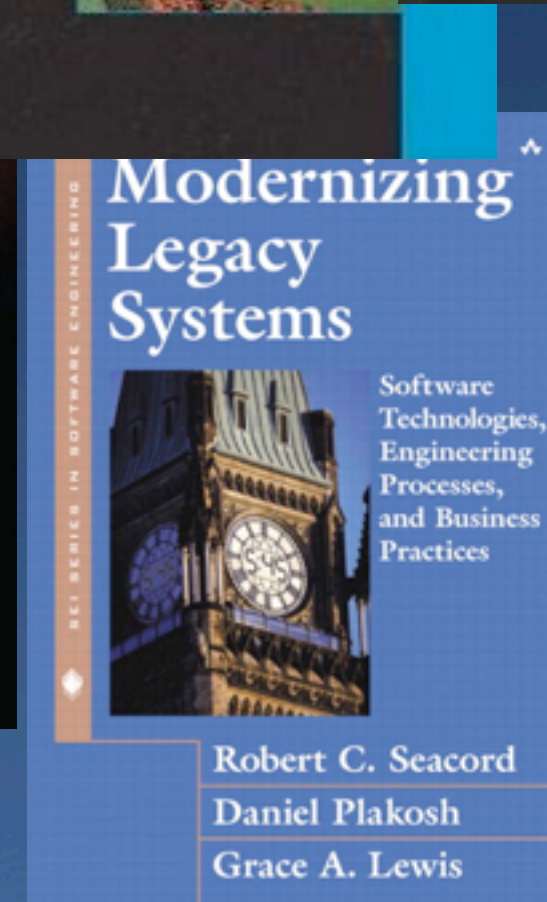
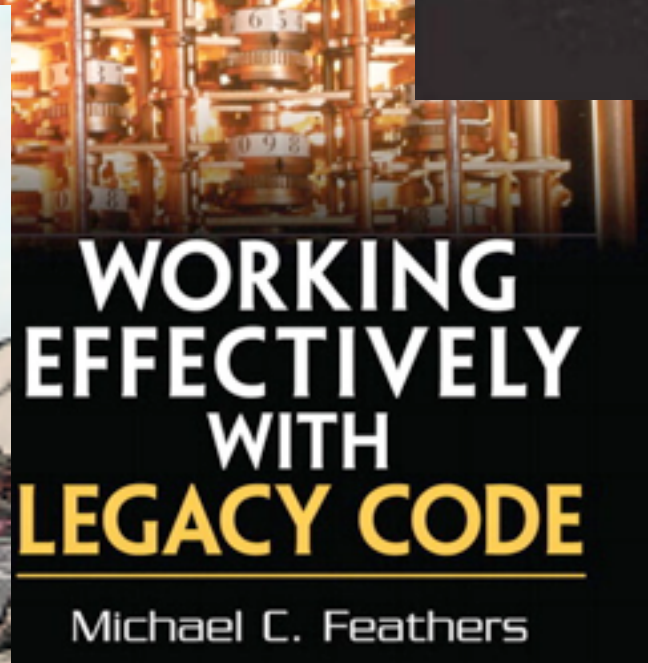
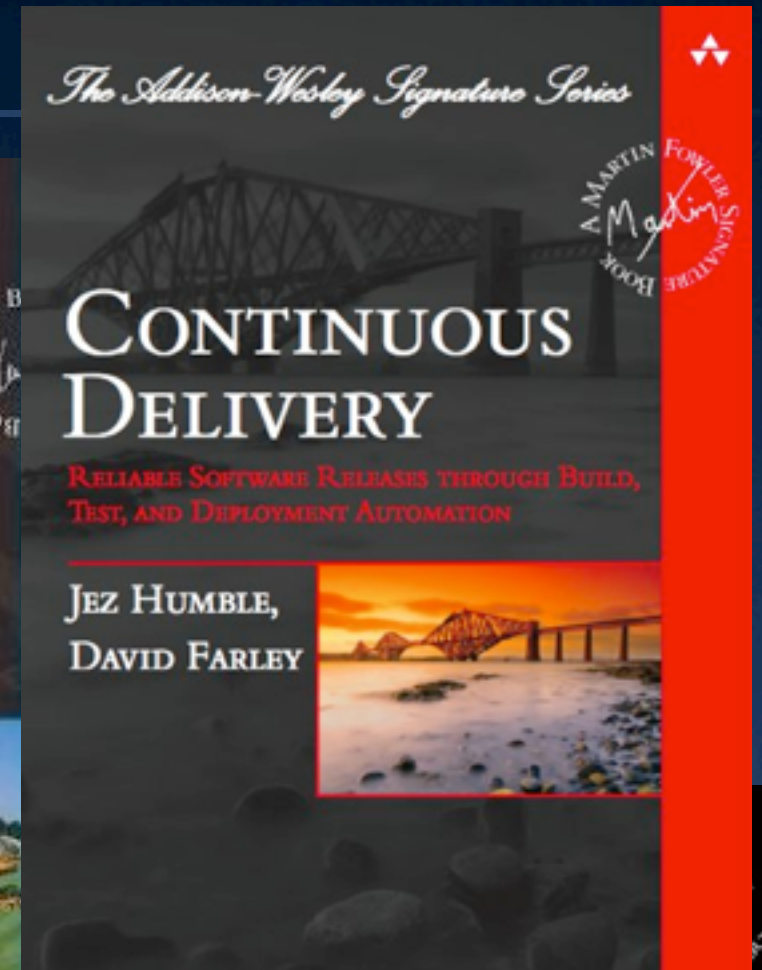
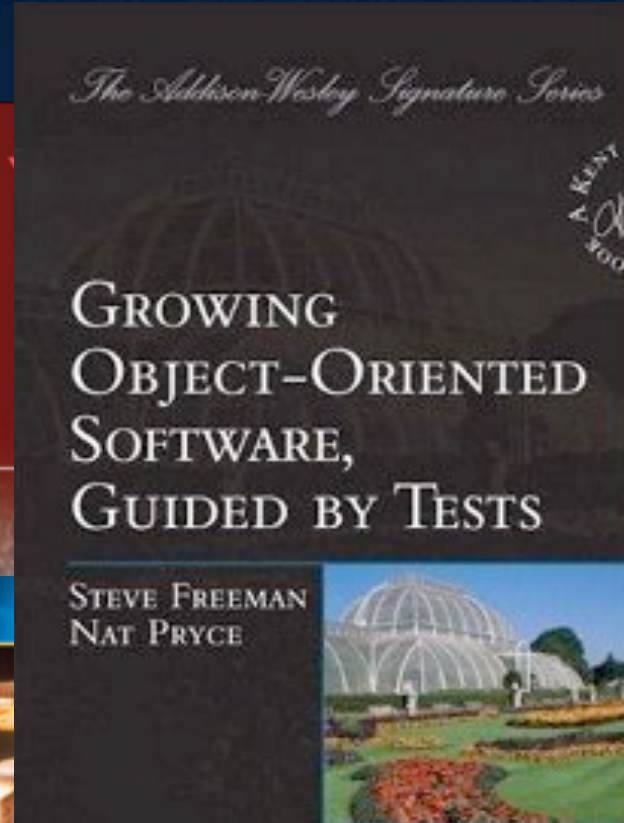
- Why?
 - in the trenches it's good to know there's a future
- How?
 - clear, simple, credible future state architecture
- For Who?
 - dev team, IT & business management
- Pitfalls?
 - timing, predicting the future

Summary

Architecture with Real Teams

- Software architecture is not just for green field projects
- Huge value in architecture techniques and principles for older or troubled projects
- Specific focus required
 - architecture techniques to find where you are
 - specific tactics for working with existing teams
- Be a master builder not in an ivory tower!

Useful Books



Questions?

Eoin Woods
www.eoinwoods.info
@eoinwoodz