QUALITY. PRODUCTIVITY. INNOVATION.

SPA 2015
28th June – 1st July 2015

Last Responsible Moment or Big Ball of Mud?

Nick Rozanski

Chris Cooper-Bland

Eoin Woods

Agenda

|	
00:00 – 00:30	**Introduction, Background and Examples**
00:30 – 00:40	**Form into Groups**
00:40 – 01:15	**Exercise 1**
01:15 – 01:30	**Break**
01:30 – 02:00	**Exercise 2**
02:00 – 02:20	**Group Presentations**
02:20 – 02:30	**Wrap-up and Lessons Learned**

Presenters

Nick Rozanski

- Head of Enterprise Architecture at ICBC Standard, London
- 30 years in IT, been working as an architect for about 15 years
- Co-author with Eoin of "Software Systems Architecture"
- Weaned on waterfall development methodologies, but now being faced with more and more agile and iterative development projects

Chris Cooper-Bland

- Looks after the Architecture discipline at Endava
- Works for clients on enterprise and solution architecture assignments, primarily in the financial services sector
- Has worked throughout the product life-cycle from systems programmer, through analysis and design to project management
- Technical interests include: architecture best practice and how this can be shared across the industry, and how enterprise architecture can be useful for developers

Eoin Woods

- CTO at Endava
- Long-time interests in software architecture and design
- Developed software using product development, waterfall, iterative and agile approaches
- Still searching for the last responsible moment!
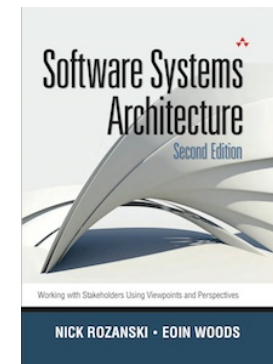
Introduction, Background and Examples
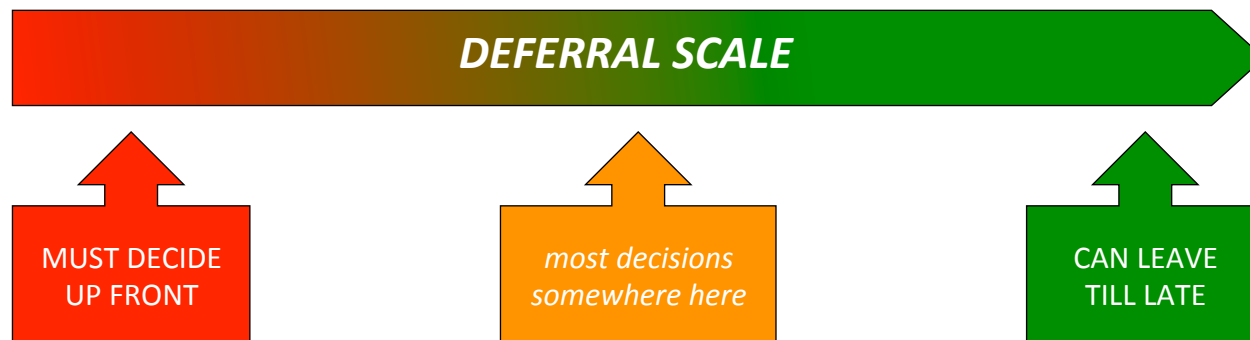
Last Responsible Moment

- In a 2000 paper for the *Lean Construction Institute*, Glenn Ballard and Todd Zabelle introduced the concept of **last responsible moment** as a foundational principle of lean project management and set based design

- They define it as:

  the point in time at which there is no longer sufficient lead time to realize [an] alternative [design]

- Lean practitioners Mary and Tom Poppendieck define it as:

  the moment at which failing to make a decision eliminates an important alternative

- In other words, once you have passed the last responsible moment to make a decision, you can't turn back

- On the other hand, before you get there, you can still change your mind if you discover there is actually a better way

The Lean Construction Institute

QUALITY. PRODUCTIVITY. INNOVATION.

Irreversibility of Decisions

- Martin Fowler also considers this idea in his (somewhat tongue-in-cheek) IEEE article from 2003, "Who Needs an Architect?"
- He makes the point that that **irreversibility of decisions** is one of the prime drivers of system complexity
- Some design decisions clearly can be deferred, such as those which affect your system's look-and-feel
- Others have to be made early on – for example, your choice of implementation stack
- However the majority of design decisions fall somewhere between **Must Decide Up Front** and **Can Leave Till Quite Late** (we call this range the **Deferral Scale**)

DEFERRAL SCALE

MUST DECIDE UP FRONT

most decisions somewhere here

CAN LEAVE TILL LATE

QUALITY. PRODUCTIVITY. INNOVATION.

Decisions, Benefits and Risks

endava

DECIDE UP FRONT	DEFERRAL SCALE	LEAVE TILL LATE

BENEFITS

- provides clarity of direction to technology teams
- provides clarity of direction to planning, budgeting and staffing functions
- mitigates long lead times for technology purchases
- mitigates dependencies with other projects and programmes
- may be necessary to meet organisational or 'political' needs, budget cycles etc.

- leads to better-informed decision making as more is known about the nature of the system
- provides opportunities to investigate and prove new or unfamiliar technologies before committing to them
- may avoid expensive mis-purchases (wrong technology, incorrectly sized etc)

RISKS

- you may have to make an expensive change of direction later
- you may miss the opportunity to take advantage of useful technologies which only become obvious as the architecture and design crystallises
- uncertainty of direction increases the overall chance of project failure

- deferral features may make the system more complex and unstable
- deferral features may make the system less performant and scalable
- the time and effort required to build these features may not return sufficient benefit
- late decisions may be due to indecisiveness rather than a deliberate strategy

7

QUALITY. PRODUCTIVITY. INNOVATION.

When Decisions Go Bad…

- Decisions which are made too early or too late can have a disastrous impact on a project

Decisions Made Too Late

- The Great Fire of London (1666)
  - London was rebuilt with major fire safety improvements

Decisions Made Too Early

- The NATS Air Traffic control system (1995)
  - System went live 7 years late and hugely over budget

Late Scope Changes

- The Swedish warship *Vasa* (1628)
  - A new deck was added after the hull had been built and the ship sank on her maiden voyage

Project Misconceived

- The NHS National Programme for IT (2010)
  - Hugely ambitious but eventually cancelled

Big Ball of Mud

- Our anti-pattern for today…