Shorten your architecture feedback loop

Eltjo Poort
GOTO Amsterdam 2022

Eltjo Poort

CGI

http://eltjopoort.nl

RCDA

Founder / owner

CGI's agile architecture approach

TOMTOM

ProRail

NXP

PhD

VU UNIVERSITY AMSTERDAM

Improving Solution Architecting Practices

Architecture

Practice Lead

Reviewer

Coach

Agile transformer

14

Peer-reviewed publications

2 Best paper awards

2016

Linda Northrop Architecture Award

Software Engineering Institute
Carnegie Mellon University

Combining Architecture with Agile working

Conflicting paradigms?

Too much **architecture** leads to…

- Late business value delivery?
- Trouble responding to change?
- Slow learning from experience?
- Wasted design effort?

Too much **agile practice** leads to…

- Ill-considered, inconsistent choices?
- Re-inventing the wheel?
- Technical debt accumulation?
- Short-lived solutions?

Benefits of combining Agile and Architecture

Architecture

- Up-front design
- Structural stability
- Standardization
- Stability
- **Risk and cost control**

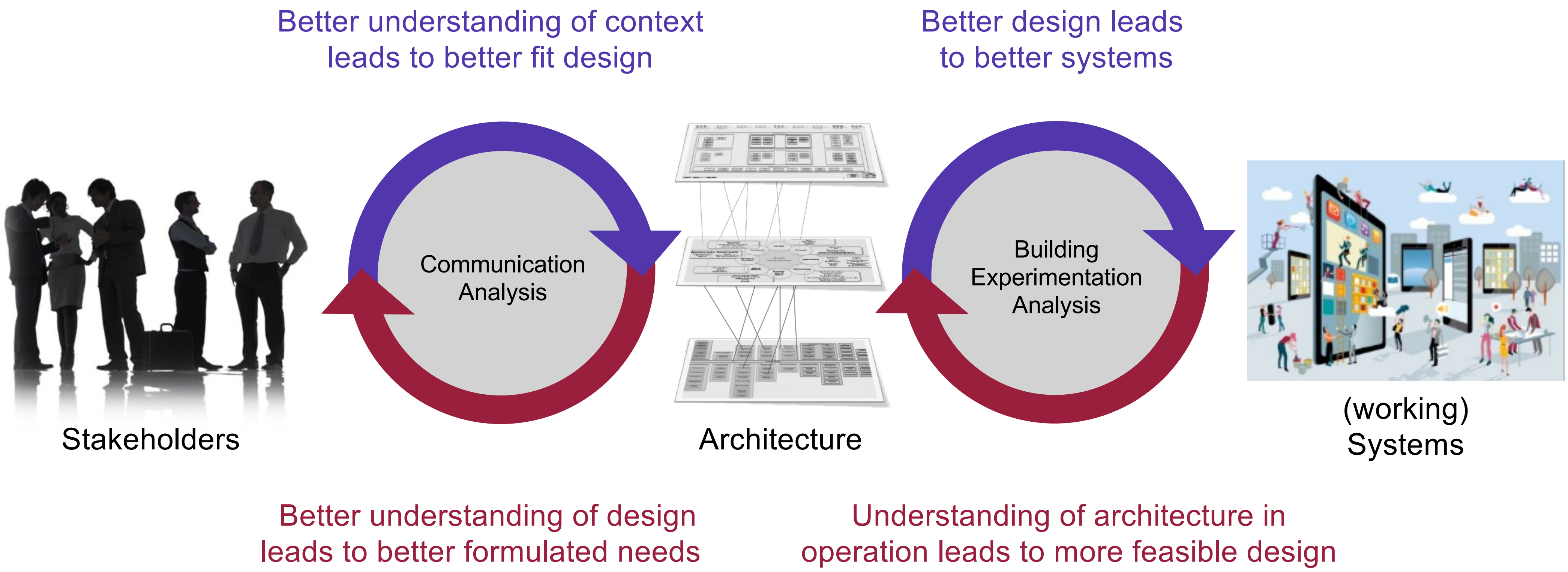
Balance

- Shortening the learning cycle
- Just enough anticipation
- Decentral where possible, standards where necessary
- Architectural design with a short feedback loop
- Balancing architecture and agile based on business rationale and not on dogma

Agile

- Experimentation
- Business features
- Local optimization
- Flexibility
- **Quick business value**

Two Architecture Feedback Loops



Better understanding of context leads to better fit design

Better design leads to better systems

Communication Analysis

Building Experimentation Analysis

Stakeholders

Architecture

(working) Systems

Better understanding of design leads to better formulated needs

Understanding of architecture in operation leads to more feasible design

The Cone of Uncertainty

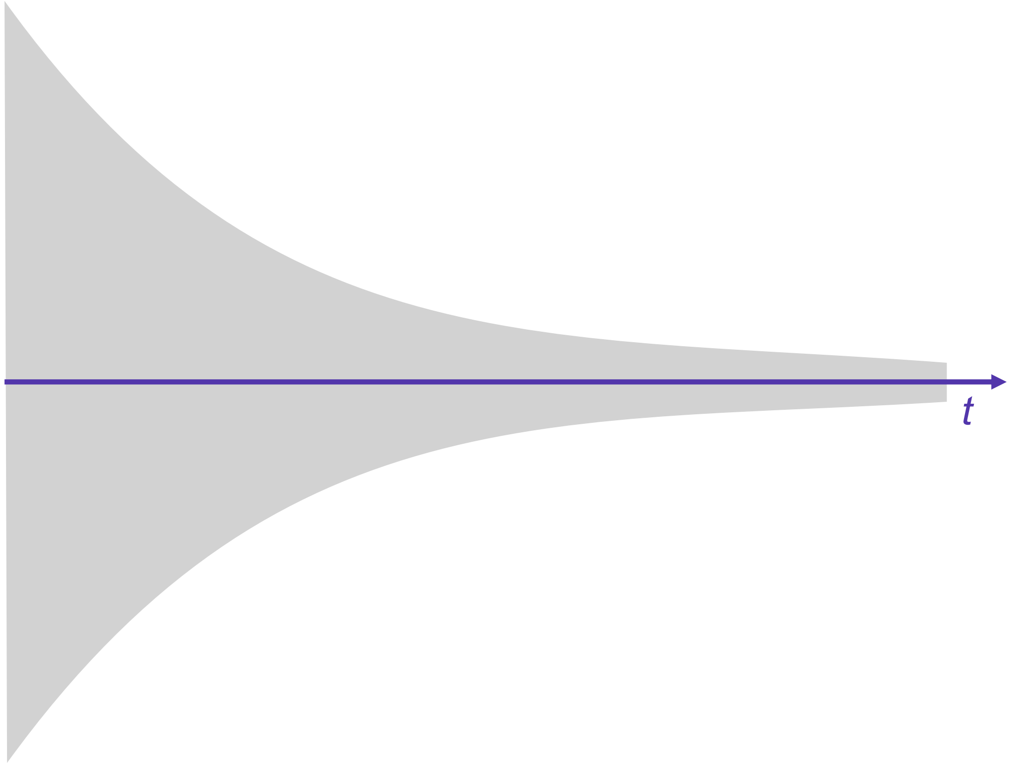
Evolution of amount of uncertainty

(in solution life cycle)

Cone is narrowed by:

- Research
- Decision making

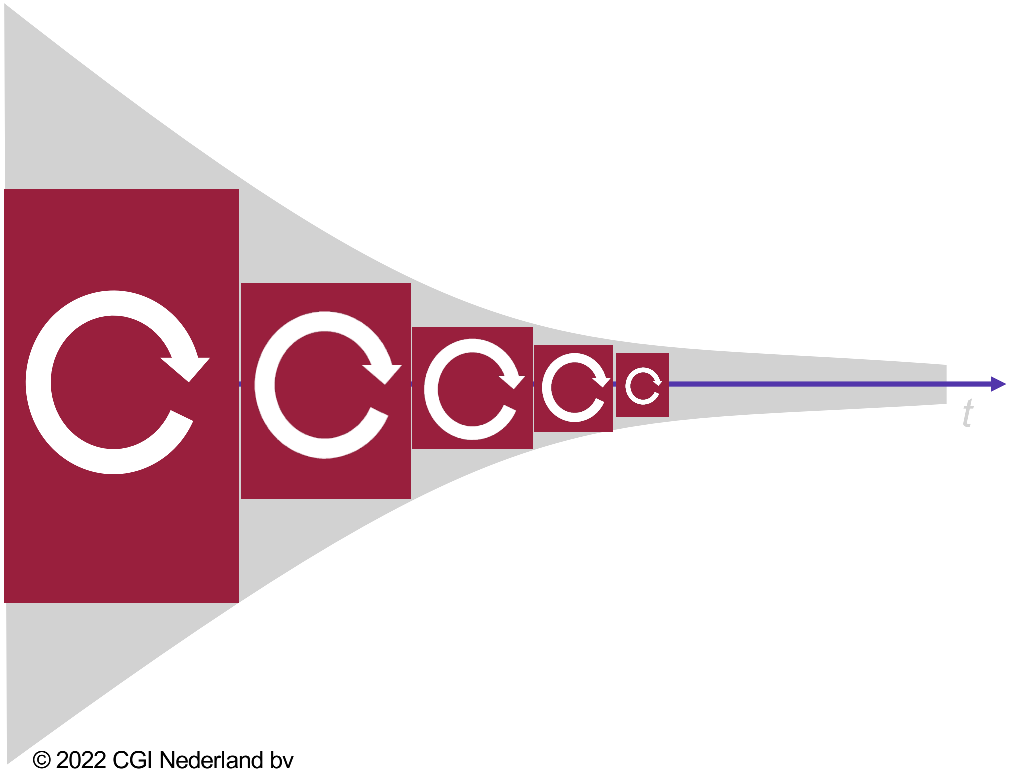
Architecture narrows the cone by researching strategies to architectural concerns and making architectural decisions.

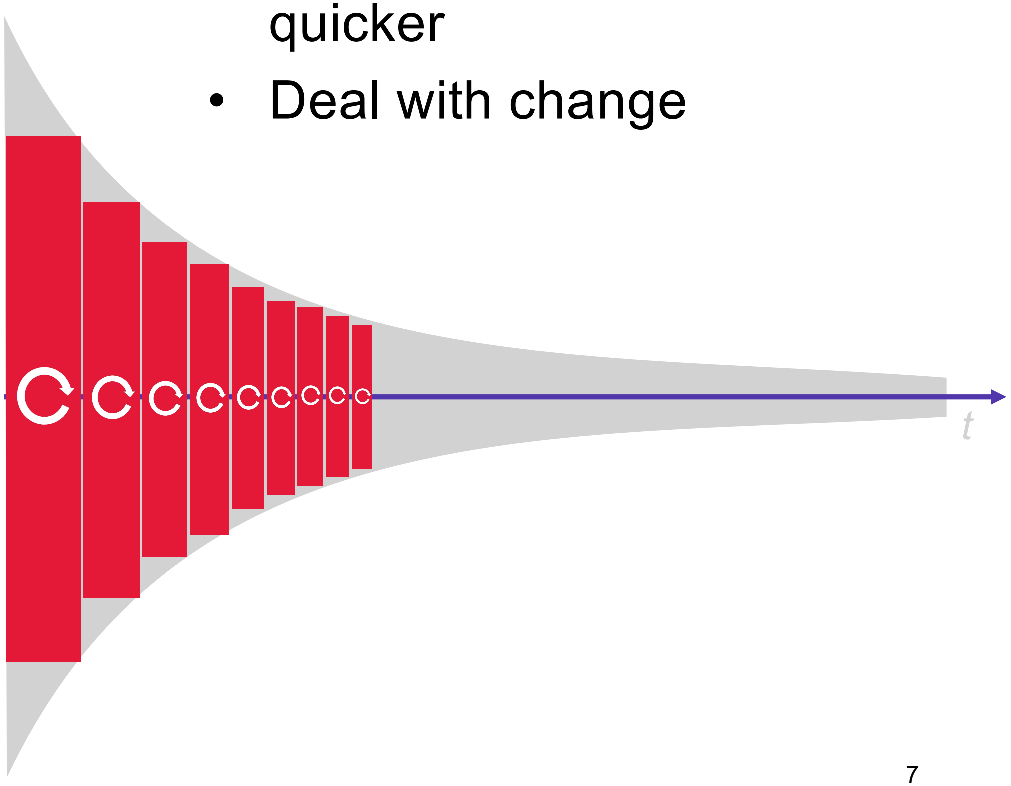


Source: Steve McConnell, Barry Boehm

Why Shorten Your Architectural Feedback Loop?

- Create better architectures quicker
- Deal with change





Tips for Shorter Feedback Loops

Architectural decisions are your primary deliverable

Continuously share concerns and decisions

Invite immediate feedback from stakeholders

Simplify your architectural documentation template

Get involved in delivery

Architectural decisions are your primary deliverable

Say goodbye to The Architecture Document (as your primary deliverable)

- Takes weeks or months to produce
- Forces approval of all decisions in one go

Say hello to The Architectural Decision (as your primary deliverable)

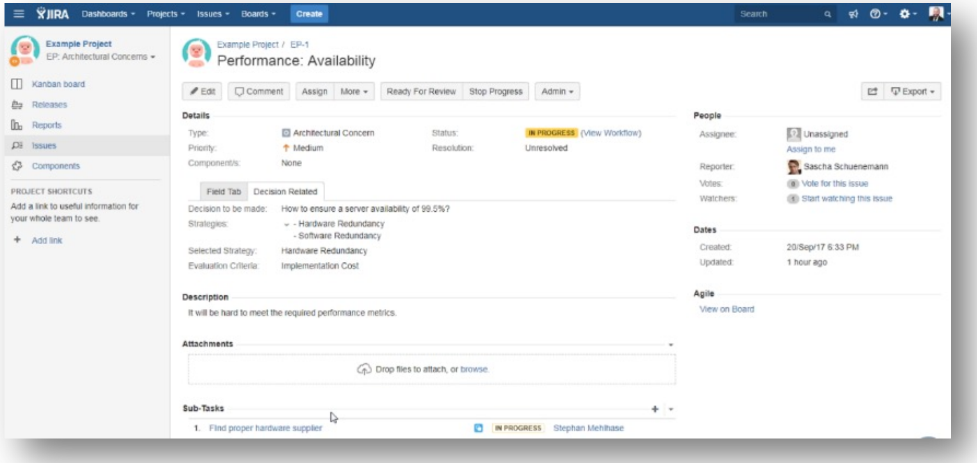
- Finer granularity of artifact
- Easier to speed up feedback cycle
- Allows individual decision timing

Continuously share concerns and decisions

Do not be afraid to share unfinished work

- The sooner you share, the faster you learn
- Share when aware decision must be taken
- Make sure decision status is clear

- Give stakeholders opportunity to contribute from the start
- Make access as easy as possible



Example Project / EP-1

Performance: Availability

Details

Type: Architectural Concern

Priority: Medium

Component/s: None

Status: IN PROGRESS (View Workflow)

Resolution: Unresolved

Decision to be made: How to ensure a server availability of 99.5%?

Strategies:
- Hardware Redundancy
- Software Redundancy

Selected Strategy: Hardware Redundancy

Evaluation Criteria: Implementation Cost

Description

It will be hard to meet the required performance metrics.

Attachments

Sub-Tasks

1. Find proper hardware supplier IN PROGRESS Stephan Mehlhase

People

Assignee: Unassigned

Assign to me

Reporter: Sascha Schuenemann

Votes: 0 Vote for this issue

Watchers: 1 Start watching this issue

Dates

Created: 20/Sep/17 6:33 PM

Updated: 1 hour ago

Agile

View on Board

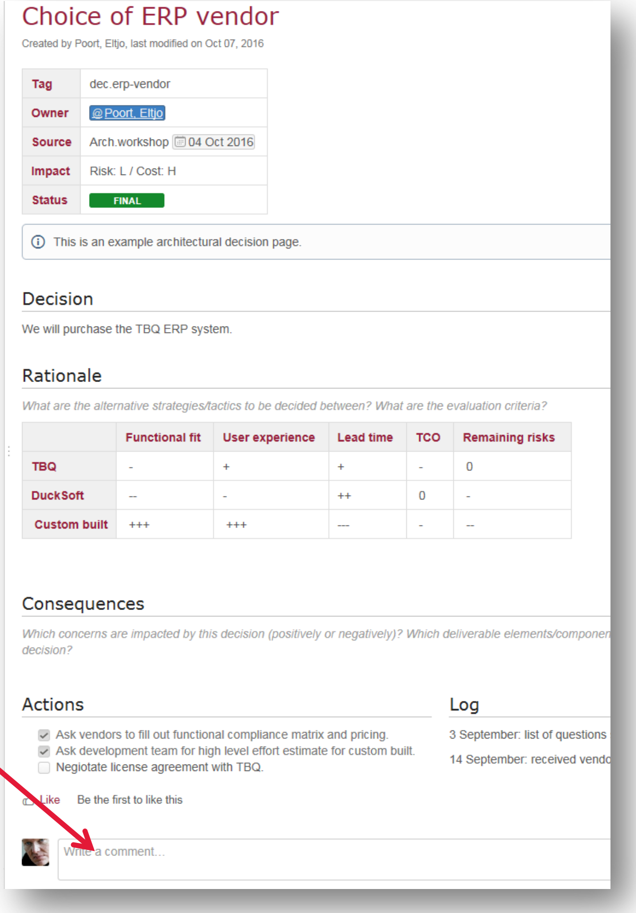
© 2022 CGI Nederland bv

Invite immediate feedback from stakeholders

- Are architects perfectionists?
- Can’t afford to wait until it’s perfect!
- Don’t wait for “official review moments” to ask for feedback (“version 0.9”)

- Make it easy!
- Put a pencil on a string next to your architecture wallpaper
- Let everyone know you welcome feedback

Most architectures emerge from a dynamic process of frequently identifying new concerns, repeatedly finding out new facts and continuously adjusting partial decisions that interact with each other and our context.



Choice of ERP vendor

Created by Poort, Eltjo, last modified on Oct 07, 2016

Tag	dec.erp-vendor
Owner	@Poort, Eltjo
Source	Arch.workshop 04 Oct 2016
Impact	Risk: L / Cost: H
Status	FINAL

This is an example architectural decision page.

Decision

We will purchase the TBQ ERP system.

Rationale

What are the alternative strategies/tactics to be decided between? What are the evaluation criteria?

	Functional fit	User experience	Lead time	TCO	Remaining risks
TBQ	-	+	+	-	0
DuckSoft	--	-	++	0	-
Custom built	+++	+++	---	-	--

Consequences

Which concerns are impacted by this decision (positively or negatively)? Which deliverable elements/componen decision?

Actions

☑ Ask vendors to fill out functional compliance matrix and pricing.
☑ Ask development team for high level effort estimate for custom built.
□ Negiotate license agreement with TBQ.

Log

3 September: list of questions

14 September: received vendo

Simplify your architecture documentation template

Template bloat kills short feedback loops

Causes of template bloat:

- Templates as knowledge repository
- “One size fits all” templates
- The Architecture Document as the (only) place to store knowledge about a solution

Fight document obesity!

Simplify your template:

- Sections for only the most common concerns at only the start of a solution’s lifecycle.
- Get rid of all ‘placeholder’ sections.
- Add views for other concerns:
  - Only as they become significant later in the lifecycle
  - Only insofar as they are relevant to your context

Create living, minimal architecture documentation.

Simplify your architecture documentation template

Separate architecture documentation goals

Goals

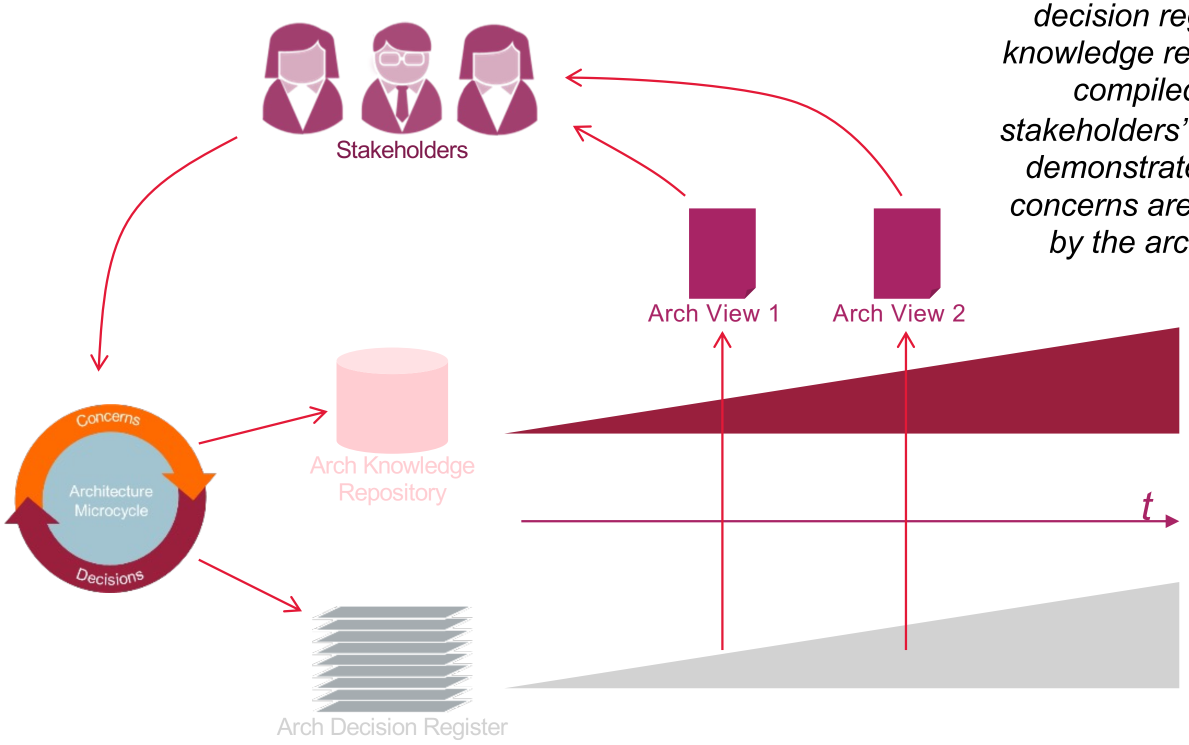
1. Evidence to stakeholders their concerns are being addressed
2. Storing architectural knowledge for later (delivery, operation)
3. Collaborate & communicate architectural decisions

Each of these goals requires its own rhythm and language

Disadvantages of “One Architecture Document”:

- Different audiences don’t speak each other’s language
- Big document delays progress and maintenance

Goal-driven architecture documentation



The architecture microcycle, driven by stakeholder concerns, feeds the knowledge repository and decisions register

When stakeholders need them, snapshots of the decision register and knowledge repository are compiled in the stakeholders' language to demonstrate that their concerns are addressed by the architecture

In the course of the solution lifecycle, the knowledge repository and decision register grow as the architecture is elaborated and maintained

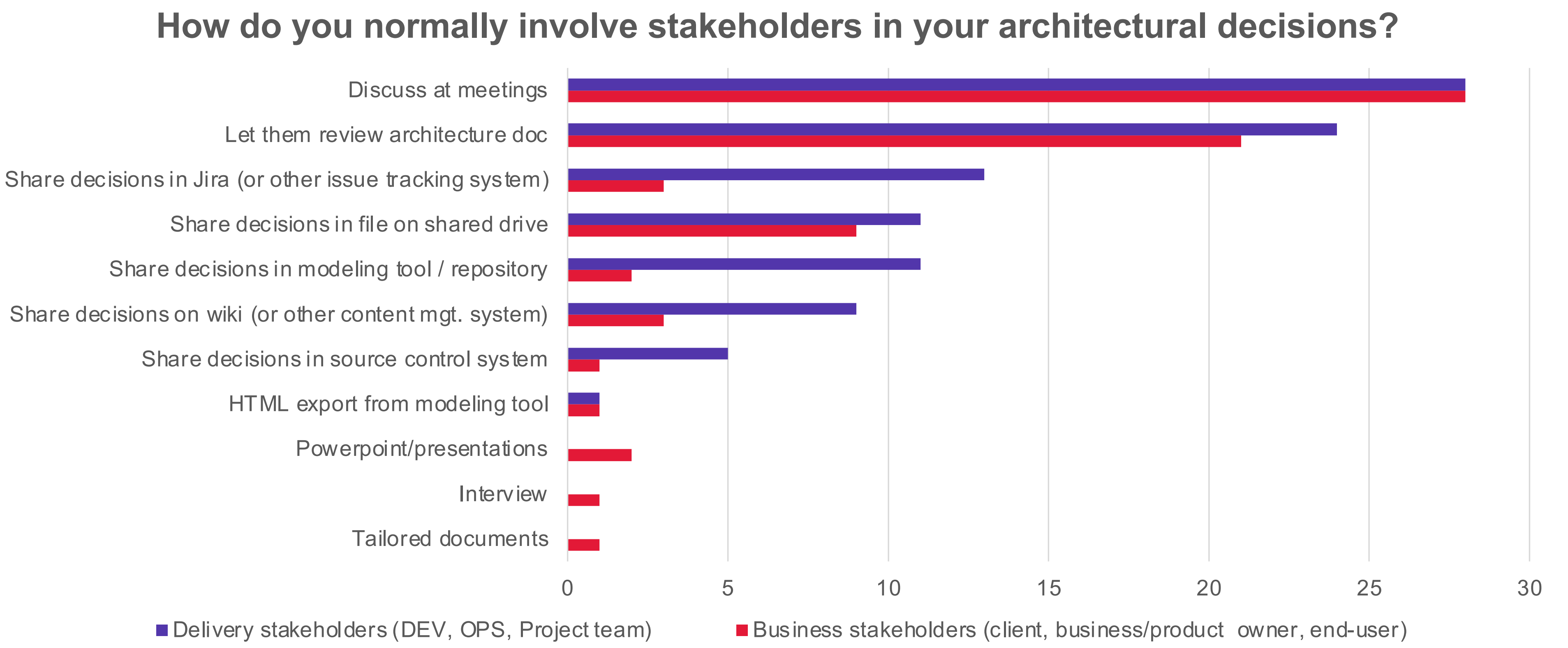
Get involved in delivery

Software architect: code key parts yourself

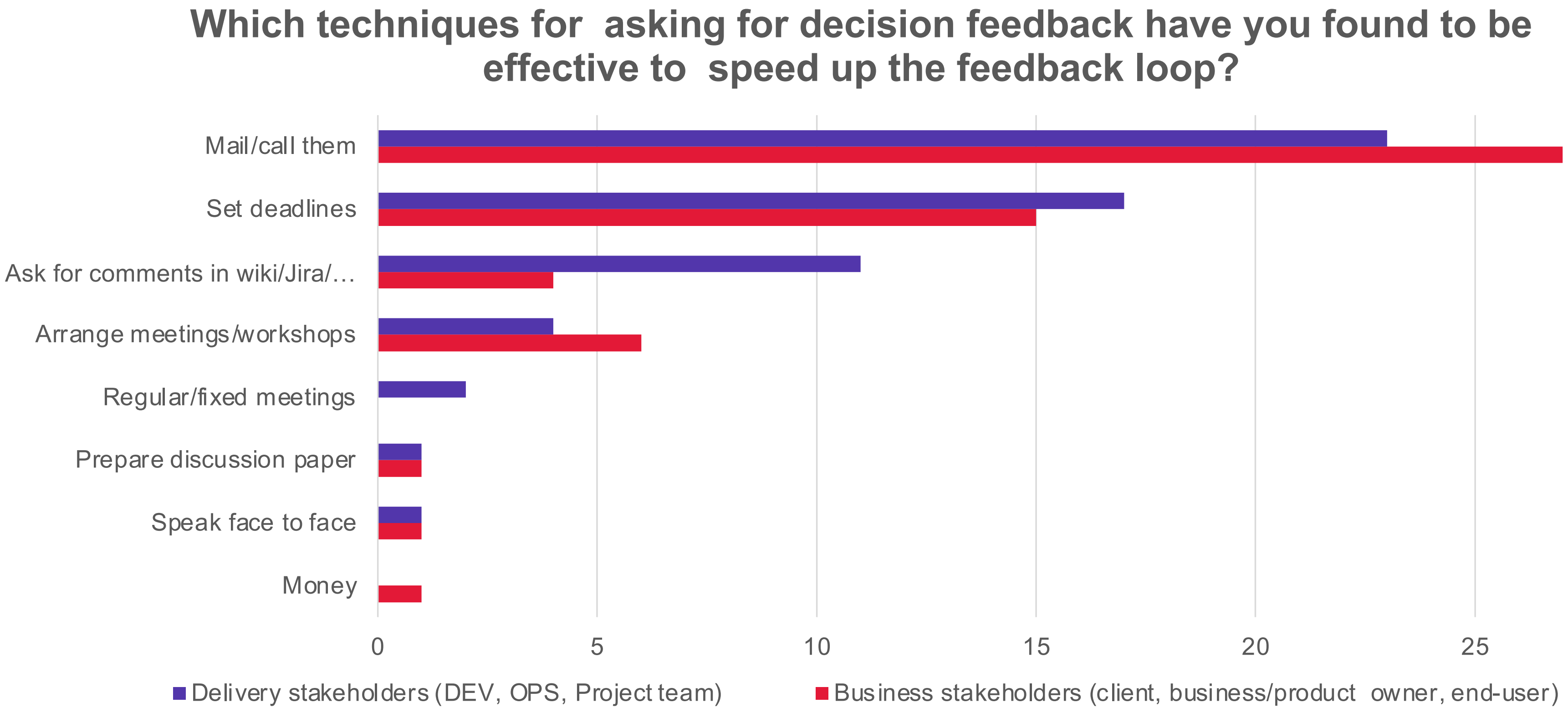
Other architect: get involved in integration, quality attribute testing or other architecturally significant delivery activities

- Become your own feedback channel
- Stimulate other delivery team members to share concerns, improve architecture

Results of quick survey about shortening feedback loop



Results of quick survey about shortening feedback loop



Results of quick survey about shortening feedback loop

“Lengthy papers slow down (or stop) decision making by business stakeholders.”

“Interestingly, our user community work better with pure text requirements than with any other form tried so far.”

“Unless dealing with other architects, it is better to focus on the *outcomes* of architectural decision rather than the technical solution itself. Focusing on the technical solution invokes strong opinions that are not helpful.”

“Politics, politics, it’s not about technology and requirements, costs versus risks, it’s about power in the organization.”

“First discuss, then write down is more accepted, educational, and more fun.”

“By only putting minimal information story in the issue tracking system and instead referring to items in the modelling repository, I ‘force’ development stakeholders to look at the model, which is much more complete. Eventually it becomes normal for developers to look at the model before coding.”

“Spoon feeding the higher-level architects with decisions and a growing number of pictures during say 8-10 weeks is preferred as opposed to writing a document in 4 weeks and then getting it reviewed.”

Conclusion

Architectural decisions are your primary deliverable

Continuously share concerns and decisions

Invite immediate feedback from stakeholders

Simplify your architecture documentation template

Get involved in delivery

Thank you

Please rate the session using the GOTO Guide app

Spare slides follow

cgi.com

Goal-driven architecture documentation

	What	Why	How (examples)	When
Arch Views	Evidence stakeholder concerns are addressed, in stakeholders' language	Collaborate with stakeholders (incl. delivery, operations) Support mgt decisions Pass project milestones	Presentations Wordy docs (Whiteboard) (Modeling tools for some stakeholders)	Project milestones Epic/story advance (On demand)
Arch Knowledge Repository	Architecture models, patterns, analysis results, research outcomes, unfinished business	Collaborate on knowledge gathering Preserve knowledge for later (analysis, design, delivery, operation)	Wiki Modeling tools Content Mgt Systems	As knowledge becomes available
Arch Decision Register	Architectural decisions, including status, rationale, consequences	Collaborate on decisions Communicate decisions Drive progress	Jira Wiki Content Mgt Systems	As soon as aware of need for decision

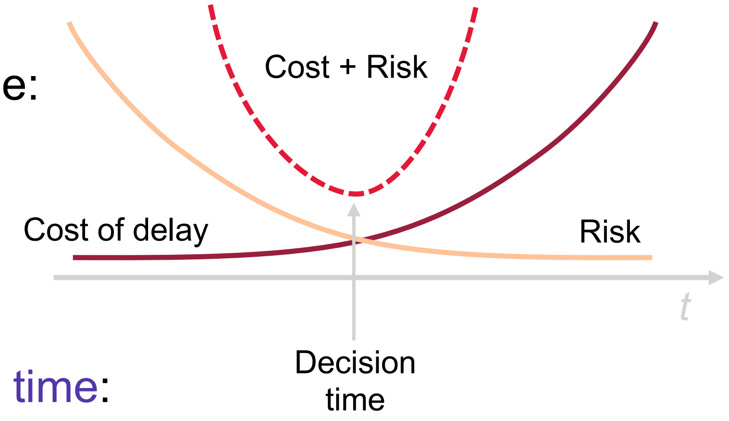
Primary deliverable

Architectual Decision Making

Timing of architectural decisions

Certainty of correct architectural decision depends on knowledge:

- relative cost of the alternative solutions
- value and impact on the business
- delivery times



Timing architectural decision is balancing risk, cost and delivery time:

- too little information → risk of not meeting key requirements
- waiting too long → project delays, wasted resources

Key skills of Solution Architect:

- timing of architectural decisions
- making decisions based on incomplete information
- dealing with the resulting risks

There's an art of knowing when.
Never try to guess.
Toast until it smokes and then
twenty seconds less.

- Pat Hein