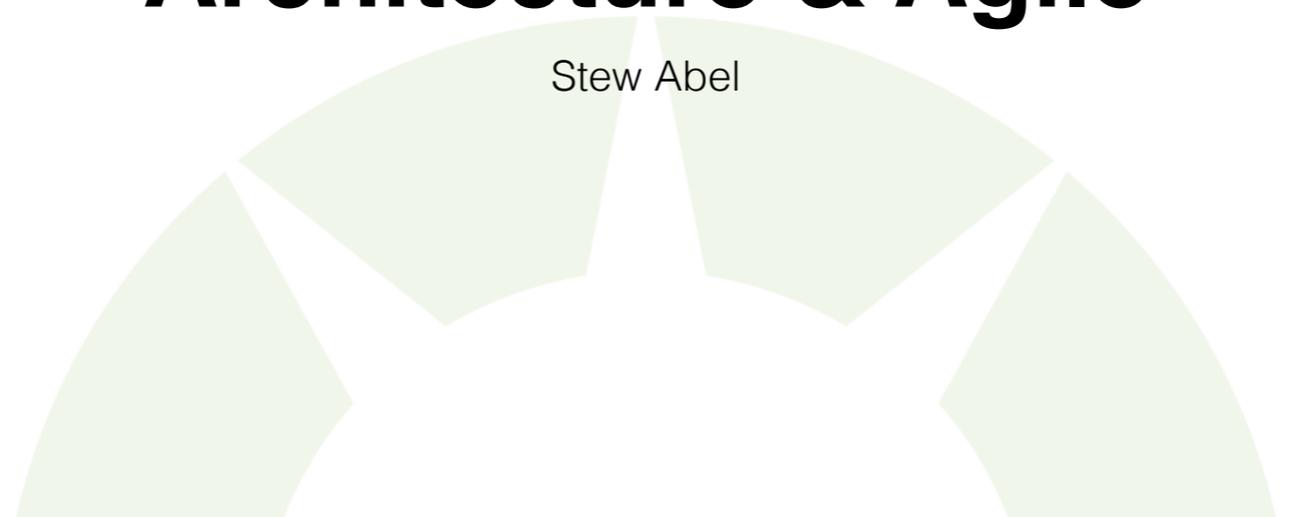


Architecture & Agile

Stew Abel

A decorative graphic consisting of several overlapping, light green, semi-circular segments arranged in a larger semi-circle at the bottom of the slide.

What is architecture?

Role not job title.
Very overloaded word.

architecture (noun)

1. The art or practice of designing and constructing buildings.
2. The complex or carefully designed structure of something.
 - 2.1. The conceptual structure and logical organisation of a computer-based system

<http://www.oxforddictionaries.com/definition/english/architecture?q=architecture>

Nothing about happening in advance.

Perhaps the idea partially comes from the older definition.



One chance at CONSTRUCTING a building

Hard to change once built

Everything needs to work together

Architecture is on display



“All architecture is design but not all design is architecture. Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change. ”

– Grady Booch

http://www.ibm.com/developerworks/community/blogs/gradybooch/entry/on_design

Worked at IBM

Developed UML

Design which is easy to change - out of the realms of architecture?

The role of architecture



The big picture

Stepping back from the minutiae of low level coding

Provide a vision and structure

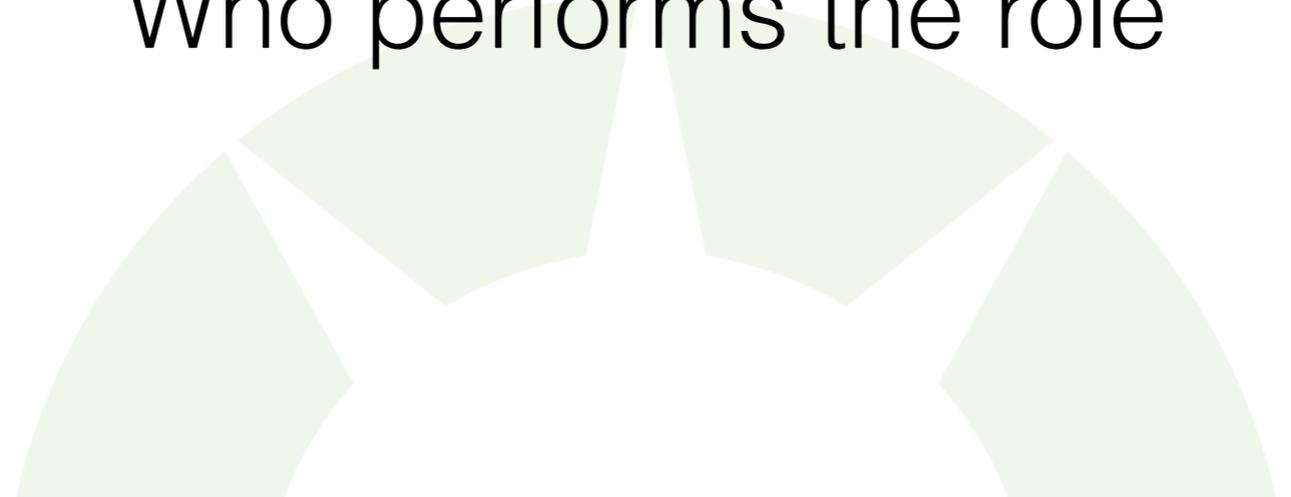
Consider constraints

applies at different levels of abstractions

'The other stuff'

- Cross cutting concerns such as auditing
- Alignment across products if required
- Micro service architectures

Who performs the role



May be someone with architect in the title

May be some or all the developers

Aim for a combination

Architect should pair with devs - understand their pain points

A shared understanding



we talk about building a shared understanding of the problem domain - leads to requirements/solutions
applies equally to the constraints & reasons for architectural decisions

Validating the architecture

A decorative graphic consisting of several overlapping, semi-circular segments in a light green color, arranged in a semi-circular arc at the bottom of the slide.

How do we know the architecture is achieving its purpose

A shared understanding improves feedback from the developers

Use metrics - from the dev process, and things such as load testing

Working software over comprehensive documentation

Manifesto for Agile Software Development

We still need documentation

The documentation needs to have a purpose

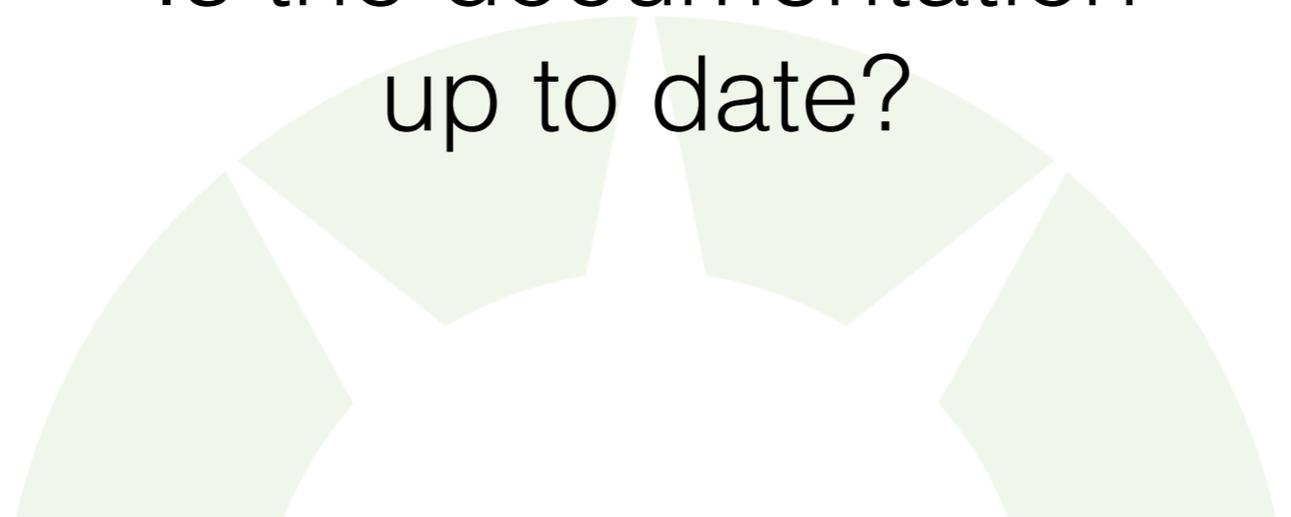
Different views for different purposes

Manage complexity

Work at different abstraction levels

Platform for a conversation

Is the documentation
up to date?



Should not become a chore

don't be afraid of deleting documentation once it has served its purpose

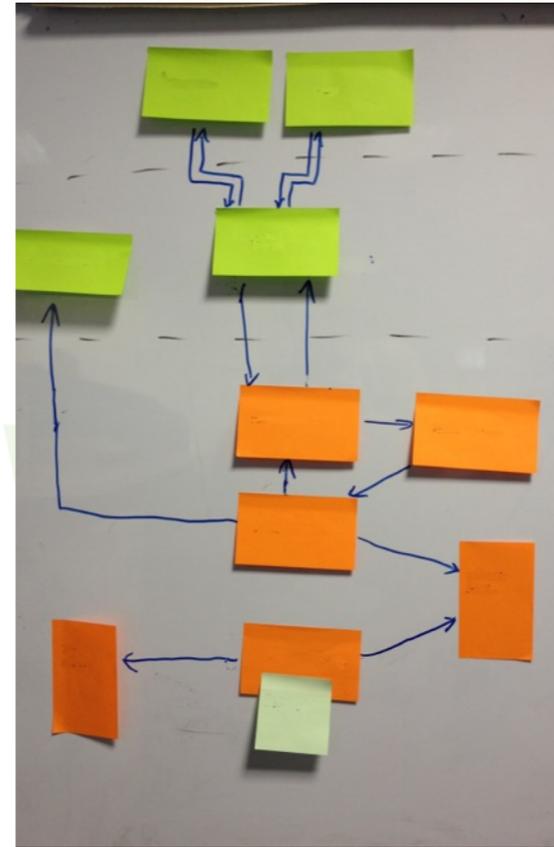
Documentation needs to live and evolve

Make architecture a
part of your process

Collaborate

Regular architectural discussions

Post It Note Design



CRC Cards - Class Responsibility Collaborators

XP concept

Could well be '**Component** responsibility collaborators'

Story checklist

Include as part of analysis of a story

Does our current architecture support this?

Defer decisions

Example - Bob Martin / Fitness

Learn more to reduce risk

@stewabel

stew@grupettosoftware.com

www.grupettosoftware.com