W4156

Agile Architecture

Wider Reveal

Given we have *so* much to cover and we also want to keep ‘layering in/piling on’ complexity I have, thus far, hidden you from ‘architecture’. In the project I have selected/encouraged the architectural pattern (web app).

In many ways this mimics the career progression of a software engineer who will initially join to write code within a platform but who will quickly take on more significant design/architecture responsibility.

Let’s now explore the architecture of software systems ……..

Agenda

Moving onto our next sub-discipline of software engineering: architecture

- ❏ Prologue: Meal Dispatch System
- ❏ Defining Architecture
- ❏ Breaking Dawn and Tribes at War
- ❏ Architecture
    a. Value of Architecture
    b. The Architecture & Communicating the Architecture
    c. Architecture Process
    d. The Architect

Prologue

Prologue: Meals Dispatch System

The residents of the Manhattania have lost the ability to cook[1] and need meals delivered. Design a system that

- Can request meals to be picked up and delivered (lets ignore ordering etc)
- Maximizes throughput. Maintain dispatch delay < 15 minutes
- Peak usage 1k requests / minute
- Restaurant logo comes up as splash logo
- Drivers, Restaurants and Diners are located throughout the city
- *Uptime* of 99.9% (9hr yearly)
- Scrappy startup has no money

Can we sketch a solution for this system?

We can assume reasonable building blocks.

[1] I was told the story of a realtor showing a manhattan studio who proudly announced it had a “fully featured kitchen” when in fact it had a single stove top and a microwave. When questioned the realtor laughed, picked up and waved a landline phone and retorted “see - fully featured kitchen”

Meals Dispatch New Requirements

You are approached by the new nation of Scotlandia.

They want to roll out as emergency medical dispatch

- Request an ambulance to take you to hospital
- Peak usage 50k calls per minute
- Request ↠ Dispatch < 60 seconds
- Roll out *nationally*
- Request ambulance is a big red button
- *Uptime* must now be 99.999% (5m yearly)
    - Must work during times of *disruption*

Can we apply the previous system? What needs to change?

What Happened?

- *Functionally* they are similar system (pick up, drop of, transport, route)
- Are we (un)comfortable using the first system for medical dispatch?
- What are the key differences between the two needs?

  (Q_____ A_____)
- Are any requirements opposing/conflicting?
- Do some of the requirements ‘matter more’ than others?

More Broadly What Did We Do?

(are we architects?)

1. We solved engineering problems to produce a design we all shared a common understanding
   a. We had some form of notation to *convey* that design
   b. We didn't design 'everything'. We chose to focus on what was 'important'

2. We followed (some) process to translate the requirements into a system design
   a. Not all requirements were of the same 'significant'. We found we could ignore a few requirements
   b. We adapted to changing requirements and evolved the architecture

3. How did we as a class produce the system design and what roles did we play?

Introducing Architecture

“The Architecture” (shared understanding / artifact)	The “important” set of decisions throughout a software project that enable 1. Ensure/derisk the project/iteration is a success / meets key Q.A 2. Captured in some artifact which effectively communicates to stakeholders [1. Architecture, Architecture vs Design & Architecturally Significant 2. views]
“Architecting” (the process)	The *process* of taking a project from idea/requirements through to architecture, construction and managing evolution of the architecture in response to evolving requirements (Requirements, [Quality Attributes], [Architecturally Significant Requirements], [Tradeoffs & Decisions], [Styles & Patterns]*, [Evolutionary Architecture], [Galls Law])
“Architect” (the responsibility)	Community organizer / engineering leader who leads this process, elevates the maturity of the team and may be responsible for trade-offs including challenging/balancing requirements.

[] everything in brackets is a key topic we will cover

* Our dear friends patterns re-appear at a higher level of abstraction for system elements and how they can relate to each other. This can be considered an *input* to the architecture process

What does Breaking Dawn have to do with architecture?

Architecture Tribes at War

	Ivory Tower Tribe	Agile and Lean Architecture Tribes	Cowboy Tribes
Philosophy	Predictive, Central planning, Big Design Up Front	Individuals, Working Software, Responding to Change, Code over Documentation (lean is different but neighbouring tribe)	Ride 'em in, cut 'em out Cut 'em out, ride 'em in, rawhide
What is Architecture?	"High level" partitioning of a system into elements and relations.	"Important" (freedom to go 'down') Flip side is we can "defer" many decisions	Whatever I develop that day
Architecture Process?	At project initiation	Iteratively / Evolutionary Weighting 'now' + 'foresight'	"Hitting requirements then recoding", "What happened?"
Who Architects?	Distinct Architect Dreaded "Review Board"	Entire team with team lead often having responsibility/accountability for outcome	Nobody?
Criticisms	Doesn't work. Architects too abstract, too removed from requirements, domain and attempting to predict.	How to handle external/slow changing dependencies. When is 'enough' architecture. How to coordinate@scale?	Use 'agile' as cover story for no architecture. Slower overall and long term sprawl of code
Derogatory names (given by other tribe)	"Archonaughts" "Ivory Tower"	(Generally, people who sit to the left or right polarize and call you a cowboy or ivory tower)	"Cowboys"

“The Architecture”

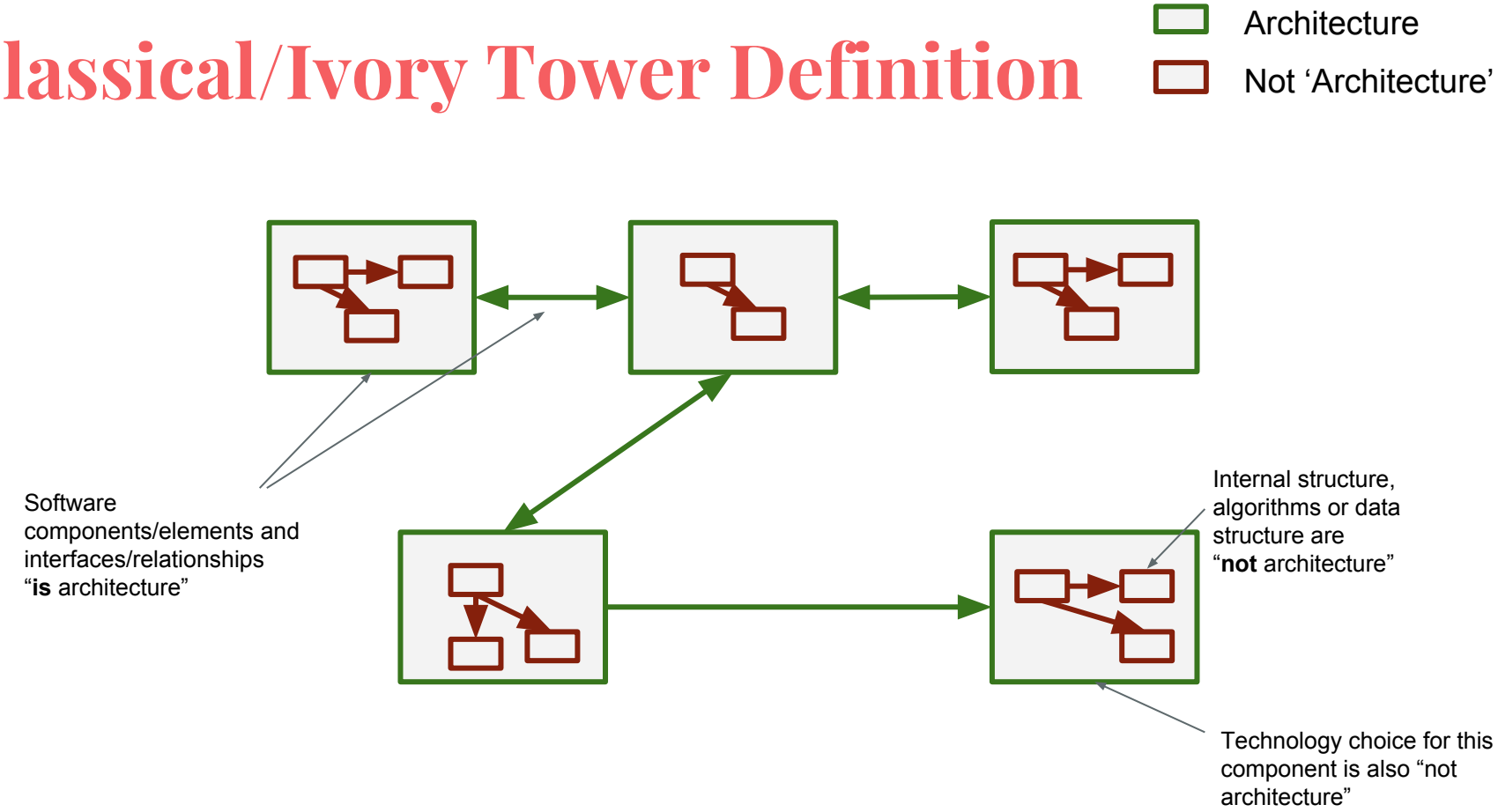
Classical/Ivory Tower Definition

architecture: *“The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution”*. [IEEE1471]

Classical/Ivory Tower Definition

architecture: *“roughly the prudent partitioning of a whole into parts, with specific relations amongst the parts” [Clements, Bachman]*

Classical/Ivory Tower Definition



I *dislike* this definition because it stratifies a system (and often roles) into ‘architecture’ and ‘design’

Architecture: Agile Definition

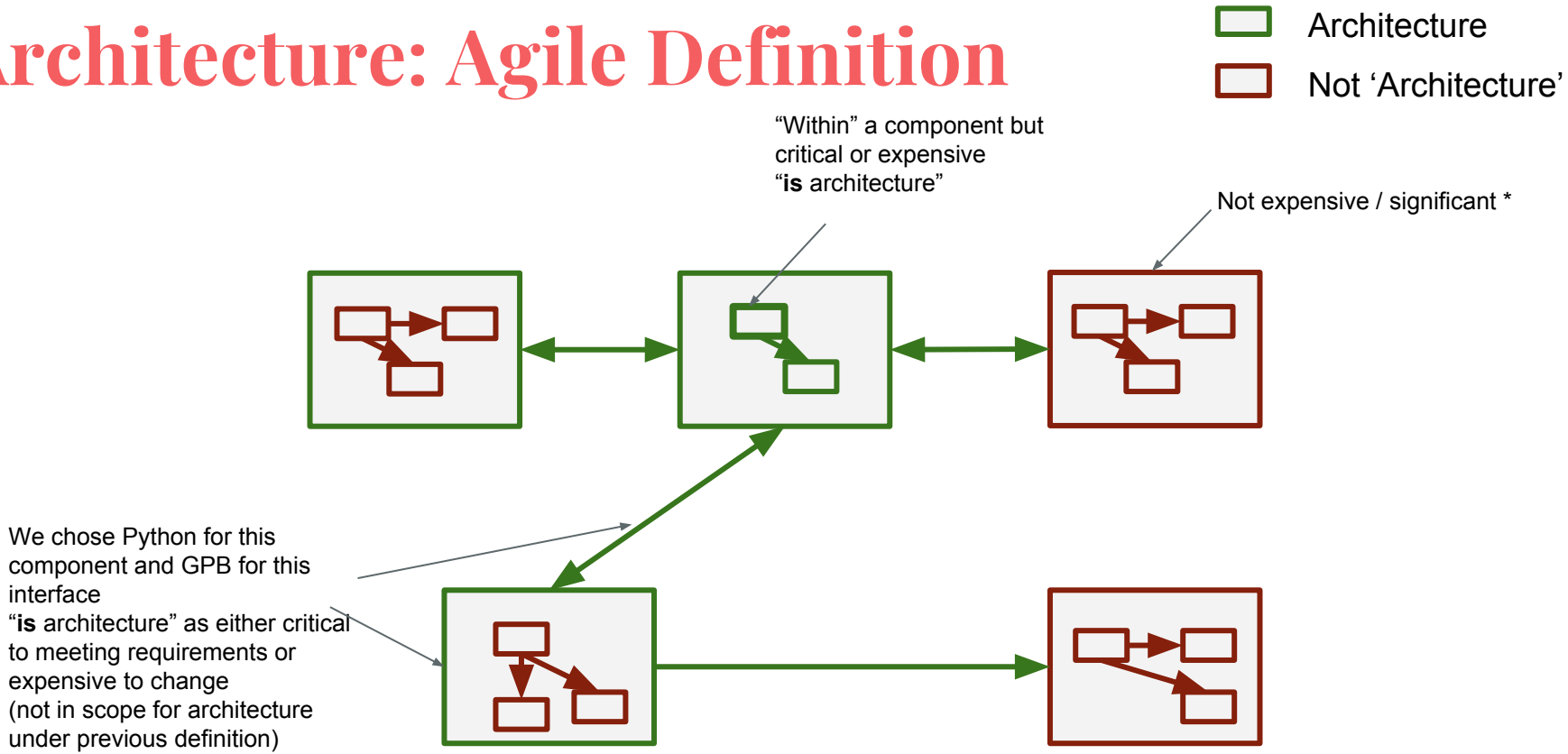
architecture: Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change [Booch 2006]

Better. But if I have a decision that is *not significant* to change then that is not architecture

Architecture:Agile Definition

architecture: *“the set of design decisions which if made incorrectly, may cause your project to fail*” [Eoin Woods, SEI 2010]

Architecture: Agile Definition



(* though when the architecture is documented all the *components* would be captured. The difference would be the amount of time spent designing this component and whether the internals would be in scope for architecture)

Architecture vs Design

“Architecture is design, but not all design is architecture”

“That is, many design decisions are left unbound by the architecture [and deferred to downstream developers]” [Clements]

Does Architecture vs Design vary?

Does ‘architectural significance’ vary based on the system?

“Why Architecture”

(Some of the following equally apply to design vs architecture activities but we are covering now in the course. Plus I argue they are especially valuable for ‘architectural’ decisions where the cost of change and criticality is higher)

Satisfy Requirements

#1 reason to spend time on architecture and design is to meet requirements. Remember we discussed the large number of project failures and requirements issues.

In architecture we have a special emphasis on quality attributes which are often determined by architectural decisions.

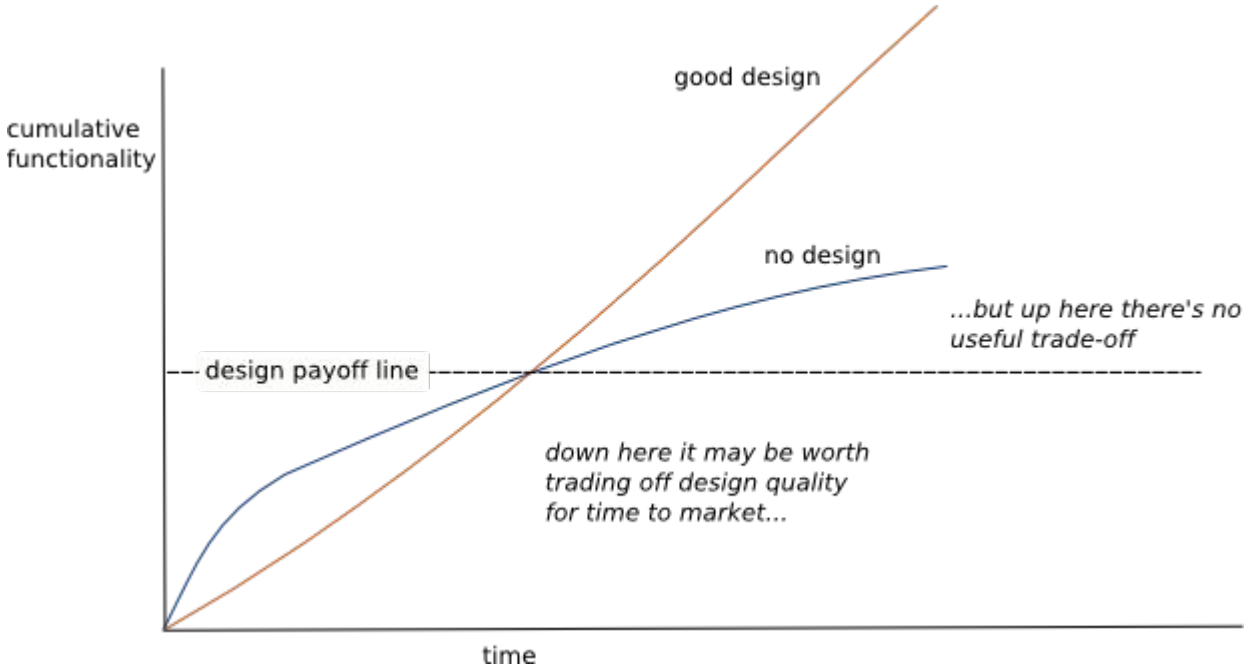
I often use a technique in project turnarounds of “what are the top 10 requirements”. You would be amazed the number of teams which had poor definition or struggled to meet top 10 requirements

Cohesion and Communication

When you have a development team or organization communicating a cohesion vision of direction accelerates build time (software engineers are stakeholders)

Often the person paying the bill wants to know what is being built (a different type of stakeholder)

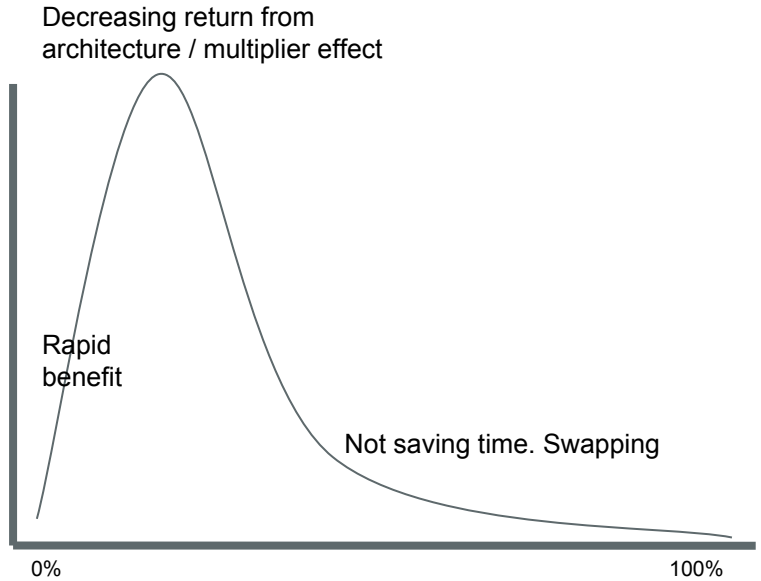
Design Stamina



https://martinfowler.com/bliki/DesignStaminaHypothesis.html

Save Time

Productivity Team
(non trivial project over medium term)

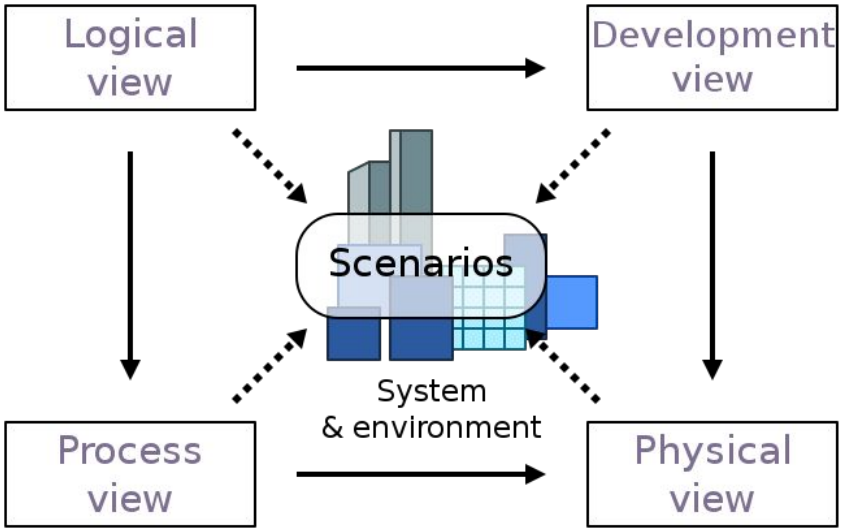


$$Development\_Time_{with\_architecture} + Architecture\_Time < Development\_Time_{without\_architecture}$$

[adapted from Clemens]

“Communicating the Architecture”

Views 4+1 (extension of the birds wing)



[Krutchen]

Pop Quiz

Question	Answer
Architecture is {the ‘high level’ layout of the system / the ‘important’ decisions to ensure the project succeeds}	
Why spend time on architecture (and design)?	
How much time should we spend on architecture (and design)?	
We can adequately describe a software system in a single view? {True/False}	

Reading

Reading	Optionality
Making architecture matter	Required
Design Stamina	Required
Technical Debt	Required
Technical Debt Quadrant	Required
4+1 Views	Required