Continuous Architecture Framework

Table of Contents

Chapter 1. Overview

By Frédéric Lé <fle@agileddd.com> and Jean-Pierre Le Cam <jean-pierre.le-cam@wanadoo.fr>

The Continuous Architecture Framework is for enterprises wishing to architect their enterprise for digital. Business leaders have long taken responsibility for creating new business models and new operating models. Too often technology is casted as an after-thought, an enabler which is not seen as a source of innovation. We argue that in a digital world, business leaders can no longer delegate technology decisions.

The old fashioned way of aligning business and IT tends to neglect the transformative potential of digital technologies. Even if it does, the waterfall nature of such business/technology interlock does not take into account the need to continuously adapt to an evolving ecosystem characterized by new competition and changing customer expectations.

The figure Dual Business/Technology Transformation advocates a continuous business/technology change journey that is led by leaders who deeply understand their business and the transformative power of digital technologies:

- Deliver a superior experience to clients and employees, and
- Provide a sustainable competitive advantage

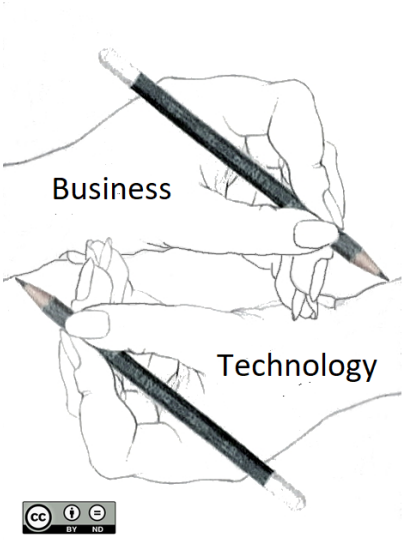


Figure 1. Dual Business/Technology Transformation

bbb

1.1. What is the structure of the framework?

vvv

1.2. How to use the framework?

zzz

Chapter 2. Rituals

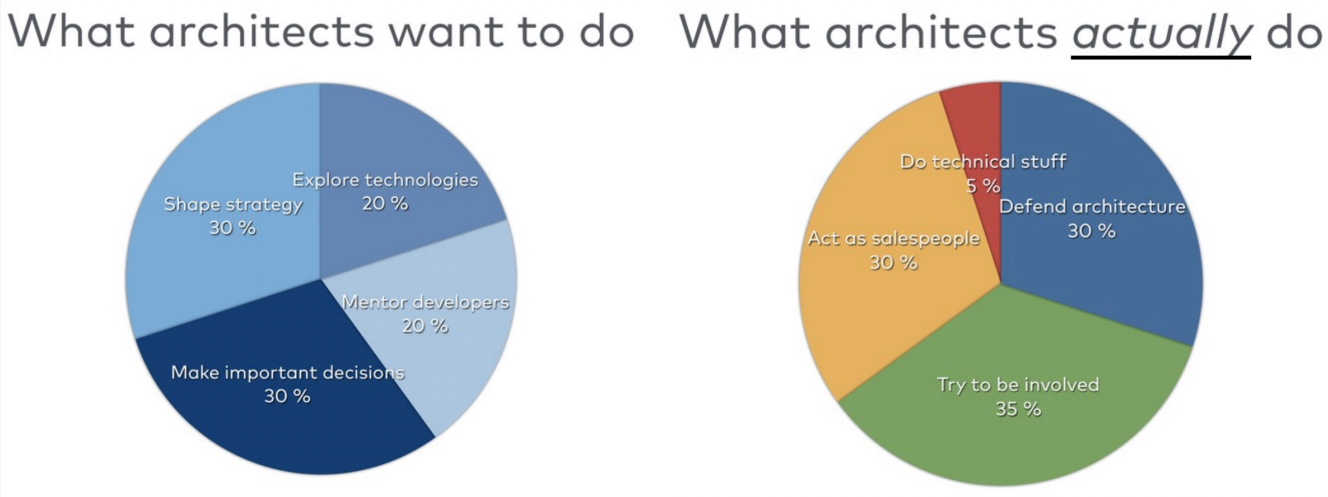
This chapter covers the Continuous Architecture rituals.

2.1. Architecture Kata

By Olivier Jauze <olivier.jauze@michelin.com> and Nicolas Chevalier <nicolas.chevalier@gluendo.com>

2.1.1. Can we help architects to practice?

Unless you're practicing architecture every day, we often spend our time on very different activites as depicted on the below charts.



(courtesy from Stefan Tilkov).

So if you find yourself in this situation, we have a ritual that was created to help you: it's the architecture kata and we want to give credits to Ted Neward who formalized it several years ago.

2.1.2. Why having a ritual to practice architecture?

The two below quotes are self explanatory we believe:

- "How do we get great designers? Great designers design, of course." --Fred Brooks
- "So how are we supposed to get great architects, if they only get the chance to architect fewer than a half-dozen times in their career?" --Ted Neward

Naming this ritual kata is a direct analogy to the Japanese Kata word (□, □) where it literally means "form referring to a detailed choreographed pattern of martial arts movements made to be practised aloneor within groups and in unison when training. It is practised in Japanese martial arts as a way to memorize and perfect the movements being executed." — Wikipedia

The idea of an architecture kata is then to repeat the architecture "movement" often enough to memorize it and perfect it over time.

2.1.3. How the ritual is conducted?

There are only two pre-requisites to organize an architecture kata: you need a topic / problem to work on and you need to have at least 2 hours ahead of you.

A kata is organized is 4 stages

1. Warm-up phase: the moderator (who can be seen as the customer) presents the topic / problem he wants the different groups to work on. He takes time to answer any clarification questions so attendees discovers the topic. Then we assemble the groups / teams. Depending on the number of persons attending the kata, we can create one or several teams of 5 people. If you have multiple teams, there will be another benefit to this practice: having several teams solving your problem, you may end up finding one solution or more ;)
2. Architect phase: each team is trying to outline an architecture that fix the given problem. And there are very few rules to respect
    - You may ask the Moderator any questions you have about the topic/problem.
    - Any technology is fair game. Customers really don't care, most of the time, what kind of technology you use. Just be prepared to defend it use during the show off Phase.
    - You may safely make assumptions about technologies you don't know well. However, any assumptions you make under this rule must be clearly defined and described during the show off phase.
    - You may not assume you have hiring/firing authority over the development team. No assumptions of developer All-Stars; assume that a development team roughly as skilled as the one you work with on a daily basis will be doing the implementation.
3. Show off phase (or peer review): During this phase, your team will be either presenting to other groups, or listening to other groups. If you are presenting, you have to explain your proposal and answer questions on it. If you are listening to the other groups, your job is to ask questions and please try to keep the questions constructive. But feel free to openly question any choice or decision that you think might not have been carefully examined or thought out. The moderator here will help to lower down the tension if a kind of back-and-forth passionate discussion starts. Remember that teams may assume anything about a technology they don't know well, so long as that assumption is clearly spelled out; if they assumed something that you know to be false, by all means inform them of that, but bear in mind that someone else in the room may have different experience with it than you, and keep an open mind. Also keep in mind that you'll be presenting your solution soon ;)
4. Vote and celebration phase: after each team has finished their presentation, we move to the voting phase. The Moderator will call out a "1-2-3", and you will each individually give the team an overall feedback vote:
    -  : You thought they nailed it, answered all obvious questions, chosen credible and feasible tech that at least seems credible and feasible, and they have a basic vision of solution. This doesn't mean they have every answer, or that they have already produced a UML diagram, it means that you have confidence they could produce them given enough time.
    -  : You thought they missed a few things, enough to make you a bit uncomfortable

that they really have a clear vision of what they're trying to build. They forgot some key questions to ask the customer, they forgot some important aspect, or they just in general didn't give you the feeling that they really "got it".

- 👎 : You thought they missed it. They made major assumptions that you think had no validity to it. They never asked the customer any questions and got burned on a few bad assumptions as a result. They really bungled the job, and you would never want to work with them on this topic.
- But all is not finished! Once the voting is complete, the ancient Klingon proverb must be heeded: "Revenge is a dish best served cold". The team departing the stage chooses the next one, and we repeat again by going back to the Vote and celebration Phase for them.

The below visual recaps how an architecture kata is conducted: