

Even self-organizing
teams must have an
architect!

...and ALL teams should have a Tech-Lead

by Erik Drolshammer and Bård Lind

What's in a name?

Tech Lead

Chief Engineer

Architect

..same responsibility

- .. I det omfanget vi nå skal snakke om
- Ett team, eller
- ett prosjekt med opp til 5-6 personer

Even self-organizing teams must have an architect!

- Hvorfor vil vi fortelle om dette?
- Terminology chaos, our clarification.
- The troublesome architects
 - ..seen from a developer's perspective
- Even agile teams needs an architect

Hvorfor?

- Observasjon av
 - prosjekter
 - organisasjonskultur
 - dyktige arkitekter
- Retrospect av hva som har fungert.
- 4 arkitekter gir minst 5 forskjellige meninger.
 - -> vi trenger noen til å føre oss samme vei.

Terminology Chaos

Tech Lead
Chief Engineer
Architect

Integration Architect

Infrastructure Architect

Coordinating Architect

Project Architect

Organizational Architect

Product Architect

Chief Architect
Enterprise Architect

System Architect
Solution Architect
Application Architect

Mange tittler, det er mange overlappende arbeidsoppgaver.
vi har ikke blitt enige om arbeidsoppgavene til disse rollene enda.

..Accept the chaos

- Hvilke oppgaver skal utføres? Tør å spør!
- Hvilken rolle skal utføre oppgaven? Tør å spør!
- Utføres oppgaven? Tør å spør!
- Utfør dine endringer.
- Våre er Architect Categorization

Architect Categorization

- Organization Architect
 - Across projects and products (Chief-, Enterprise-, System-, Integration- and Infrastructure architects)
 - Focus close to business organization.
- Project Architect
 - Work on a single project, and within project scope. (Solution-, Software architects, Chief Engineer or Tech Lead)
- Product Architect
 - Full product lifeline scope (Chief Product-, Product-, System-, Sub-system-, Build architect or Tech Lead)

Terminology Chaos

- Tør å spør
 - deg selv og de du jobber med
- Prøv å definer hvilke oppgaver som må løses
 - og av hvem
- Avgrens til ett bestemt scope.

Even self-organizing teams must have an architect!

- Hvorfor vil vi fortelle om dette?
- Terminology chaos, our clarification.
- **The troublesome architects**
 - ..seen from a developer's perspective
- Even agile teams needs an architect

The troublesome Architects

- Før var prosjektlederen skyteskiven.
 - Tror PL er blitt bedre, for nå
- Skylder vi på Arkitekten

Eller trenger vi PL mindre gjennom Agile metoder? Er PL sine oppgaver blitt enklere i hyppige leveranser? Det er i såfall bra!

The cooperation between developers and architects is not always good. We will now take a quick look at some of the common misunderstandings and mistakes and try to give some advice on how to proceed.

The Architect

- Utilgjengelig? Sitter i møter.
- Legger regler som vi ikke forstår.
- Føler ikke smerten av sine beslutninger.
- Snakker et språk vi ikke forstår?
- Utvikler vil så gjerne formidle erfaring tilbake.

* Age gap: young developers, middle-aged architects

* Sort of a language barrier; (organization) architects and hard-core programmers use different domain languages

* Agile vs waterfall, RUP

* Powerpoint vs working code

* Heavyweight vs lightweight

sitter i Ivory Tower

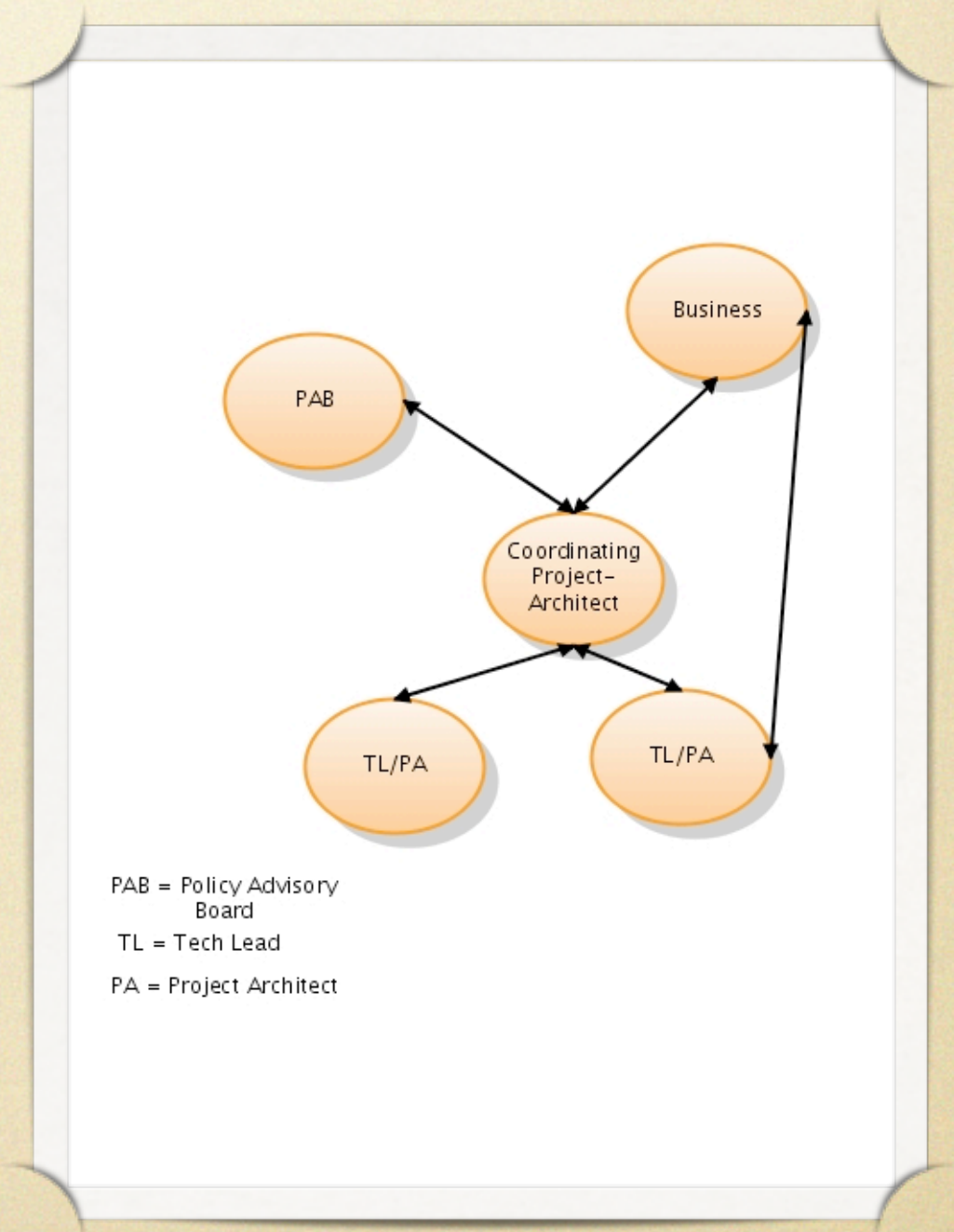
- Må forstå situasjonen utover “nå bildet”.
- Krever kontakt med flere enn ett enkelt team.



Fortell om hierarki, avensment for dyktige utviklere (ref tidligere IASA diskusjoner)
Må også ha kontakt med organisasjonen – og tiden er kun 40t i uken -> Løsning: skap en lærende organisasjon.

eller i Learning Organization

- Coordinating Architect
- Forstår situasjonen i flere teams, og i organisasjonen.
- Får tilbakemeldinger på sine beslutninger.



Skap en kultur basert på at vi skal lære av hverandre, vokse på våre feil.
Tør å spør, det er ikke nødvendig at "The Architect" skal ha alle svarene. Men skal ha erfaring til å se konsekvens av de svarene som kommer.

Forklar PAB: "Arkitekturforum" – beslutninger på regler og tekniske føringer.

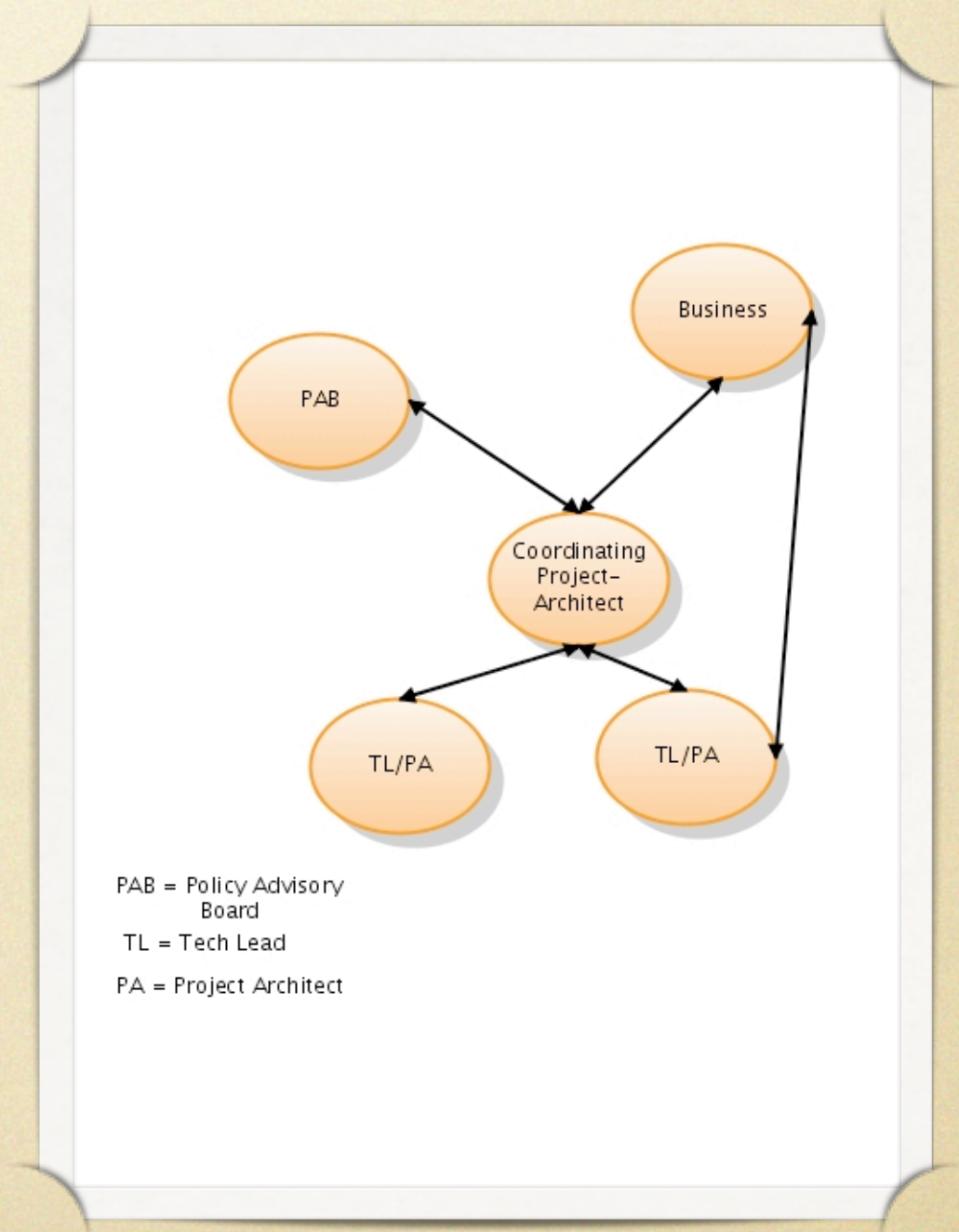
the gap

- Technology related policies
- The architect is too far from the team
- Hard to give feedback to the architect
- “Must” make the discussion alone.
 - I’m the most experienced, and must know all the answers. (aka control freak, dictator, super hero)

Arkitekten tar tak i de tingene som han kjenner best. Da får du -> technology related policies.

Bridging the gap

- Høre, lære, forbedre
- Continuous feedback
- Learning circle
- Vi må spille hverandre gode.



Learning Circle:

Mer i neste seksjon "Event Agile teams need a architect"

Lag en lærende organisasjon rundt deg selv. Det er kult med "Tør å spør". La folk få muligheten til å

blomstre. Gjødsele gode diskusjoner -> og dokumenter disse. Referat is King -> det er den felles forståelse vi er blitt enige om.

La diskusjoner være pain-driven. Diskusjoner som "dette skulle vi helst ha gjort" hører retrospect til.

Avslutt diskusjoner vi er blitt enige om tidligere, eller stop diskusjoner som ikke skal løse et eksisterende problem.

Utviklers råd til arkitekten

- Vær ikke fremmed for hverandre. Tør å spør.
- Vær tilgjengelig
 - Gjem deg ikke bort i møter.
- Vis ansiktet ditt (husk en arkitekt har 80% av tiden sin som kommunikasjon)

80-20 reglen i egen lyntale.

Even self-organizing teams must have an architect!

- Hvorfor vil vi fortelle om dette?
- Terminology chaos, our clarification.
- The troublesome architects
 - ..seen from a developer's perspective
- **Even agile teams needs an architect**

Hva jeg håper å ha vist så langt er:
Vi må bli bevist på oppgaver, og hvilke roller som skal utføre disse. Noen må ha ansvar for disse oppgavene, og mulighet til å løse dem.
Vi trenger god kommunikasjon mellom utviklere, arkitekter, business og PAB.
Siden kommunikasjon er en viktig jobb bør vi sørge for at den utføres. Faren med et selv-organisierende team er at ingen vil ta denne kommunikasjonsjobben, fordi tech er artigere.
Andre grunner vi skal se på i neste seksjon er å sikre samordnet utvikling, å holde utviklermomentet oppe -> sikre at utvikling går så raskt som vi ønsker.

Architect in self-organizing teams

- Tech - Lead == Architect i dette scopet.
- Tittelen er uviktig.
- Vi har oppgaver som en person må utføre.

Vi har nå sett på en del generelle problemer. Vi har sett på to tiltak for å få bukt med disse:

- * tydeliggjøre roller, ansvar og kommunikasjonskanaler
- * legge til rette for feedback og læring

Som en avslutning vil vi gjerne gå inn på arkitektens rolle i smidige team.

JA, også smidige team trenger en arkitekt. Det betyr ikke at man trenger en ha en person med sjefsarkitekt som stilling på hvert eneste team. Det betyr at hvert team bør ha én person som er ansvarlig for å ta valg som påvirker arkitektur og design. Denne rollen er innenfor project scope og kalles ofte tech lead, men det er ikke noe i veien for å bruke f.eks. Chief Engineer som Olve har snakket om.

Savner også... i den agile metodikken noa a-la

* prosjektleder er redusert til prosjektsektretær

* teamet skal ta valgene...

** men hvordan får vi konsistente og balanserte beslutninger...

*** er ikke det noe som pleide å være arkitektens ansvar?

**** dette ansvaret er jo ennå viktigere i smidige prosjekter, siden vi ikke har PL

==> gjeninnføre bevisstheten og ansvaret - hva du kaller "dyret" er mere en smakssak..

Oppgaver til Arcitect

- Støtte teamet i felles designansvar
 - “Lukte” problemer, få de fikset.
- Beslutningsansvar
 - Sørg for konsistente design beslutninger.
 - Sørg for beslutninger når teamet har disens.
- Brøyte veg for teamet.
 - Se etter hvilke grep som skal ligge 2-3 steg forran.
- Oversikt over kundens totale behov.
 - Semi-dealjert nivå.

Sikre fremdriften

- Sikre at eksterne avhengigheter identifiseres, følges opp og løses.
- Sikre at vanskelige temaer tas opp, forfølges og løses.
 - Høre på summingen i prosjektet, lytt etter om noe skurrer.
 - Prosjektleder kan godt være utøvende på denne.
- Sikre fremdrift
 - teknologi ikke hindrer fremdrift
 - teknologiske beslutninger ikke stopper fremdriften

Tilrettelegge for tekniske spørsmål. Sikre at teknologi ikke hindrer teamet å nå målene.
legge hodet på blokka for at det TAS valg selvom man ikke vet alt... og container eventuell skade... men legger hodet på blokka)

Oversikt

- Se kundens totale behov.
- Oversikt over hvem som gjør hvilke oppgaver
 - Støtte for PL og eksterne som trenger hjelp.
- Ha oversiktsbildet for å kunne svare på spørsmål som:
 - hvem i teamet kan løse en gitt oppgave.
 - hvem i teamet kjenner denne Use-Casen best.
 - hvem hos kunden kan utvikleren gå til for å få svar på sine sp.mål?

Utfordre

- Utvikler skal utfordre Tech-lead!
 - når beslutninger ikke er fornuftige, eller reduserer farten på teamet.
- Tech-lead skal utfordre regler og føringer fra PAB!
 - Når disse er i vegen for at teamet skal kunne levere.
 - Kontaktperson mot Coordinating Architect.

Bruker Tech-lead for å tydeliggjøre at dette er innenfor team-scope.

Det er **ikke Organisasjons arkitekter** som skal spesifisere **hvordan** et team skal utvikle, det er TECH LEAD som skal være smøringen og erfaringen som gjør at teamet produserer optimalt innenfor de rammene PAB gir.

Kontaktperson inn til arkitekturteamet gjennom Coordinating Architect.

Kontakt med Arkitektur-team

- Hvilke valg bør diskuteres med coordinating architect?
 - Ønskede brudd på eksisterende arkitektur.
 - Regler som forsinker utvikling.
 - Implementasjon i eksterne (for teamet) systemer, som gir dårligere løsning i dette prosjektet.
 - Tilbakemeldinger på ting dette prosjektet gjør bra.
- Utfordre PAB (Policy Advisory Board)

Kontakt med PAB

- Hvilke valg må tas opp i PAB?
 - Se Cantara wiki for Policy Advisory Board
- Eksempler
 - Brudd mot "Alle tjenester skal ha aktivitetslogg."
 - Ønsker brudd mot: "Alle tjenester skal ha accesskontroll mot AD."

Participants

- * IT Manager
- * Company Chief Architect
- * Service Owner
- * Project Lead Architect
- * Central PAB-issue stakeholders

PAB Meeting Agenda

- * Intro PAB
- * Presentation of the PAB issue
- * Stakeholder analysis

Even self-organizing teams must have an architect!

- Shared our experiences.
- Terminology clarification.
- Architects, seen from a developer's perspective
- Tasks to fulfill.

Even self-organizing
teams must have an
architect!

...and ALL teams should have a Tech-Lead

speaker Bård Lind
bard.lind@gmail.com
twitter: baardl