



UiT Norges arktiske universitet

Rammeverk for systemutvikling ved UiT

Johnny Hansen og Magnus Lyngra

Bakgrunn og hensikt

Ha fornuftige rammer og avgrensninger

Ha fleksibilitet nok til å fange nye trender og teknologi

Tilrettelegging for god produktivitet

Tydeligere for rekruttering og innleie

For utviklere ved SUA-Digitalisering, ledere og prosjektledere

Rammeverk systemutvikling

Standarder når man kan velge:

- .NET (C#)
- Django, Flask (Python)
- JavaScript ved behov for mer avansert UI

Dokumentasjon:

Utvikler har ansvaret for:

- Kodedokumentasjon, infrastruktur, deployment: GitHub
- Følge anbefalt praksis ihht: <https://github.com/UiT-ITA/Utviklere-Anbefalt-praksis>
- API-dokumentasjon i Swagger annotations
- Eksponerte grensesnitt/informasjonsmodell

Digital suverenitet:

Utviklerne skal

- Forsøke etter beste evne å velge løsninger og teknologier som understøtter behovet for digital suverenitet (fortrinnsvis europeiske løsninger)

Integrasjoner:

- Vi følger IntArk-prinsippene så langt det lar seg gjøre, se [God praksis i fellestjenesten for Datadeling | Sikt teknisk dokumentasjon](#)
- Gravitee som API Gateway
- RabbitMQ for meldinger
- Lenke til tjenestekort i TOPdesk SSP
- Applikasjonsmonitorering i produksjon
- Sky før kjeller

Prinsipper for hosting

- Hovedregel: Cloud native PaaS hovedretning på nyutvikling, f.eks. serverless med azure functions/AWS Lambda.
- Lokal server kun ved spesielle tilfeller, for eksempel om applikasjonen trenger tilgang til FS-databasen. Settes primært opp som docker-container.
- Skal ha applikasjonsmonitorering, gjerne Application Insights.
- Skyløsning bestilles til SDF-Forskningsplattform, se word-mal i Sharepoint .

Dersom andre standarder, plattformer eller prinsipper brukes:

- Beskrive hvorfor og avklar aksept.

Prosessbeskrivelse for utviklingsprosjekter

Sjekkliste før programmering starter:

- Arbeid delegeres via tjenesteteamkoordinator
- Utviklerteam (*to utviklere på større prosjekt*) velges basert på arbeid i kø og kompetanse, kombinerer alltid junior og senior
- Tjenesteteamkoordinator er prosjektkoordinator med mindre prosjektet vurderes av porteføljestyret, da utpeker de en (*prosjektkoordinator = driver prosjektet fremover, kontakt med prosjekteier, kontakt mot brukergrupper*)
- Løsningsforslag og brukerhistorier i hovedsak avklart med bestiller
- Systemarkitektur avklares med arkitektteam.
- Samarbeidsform, møtestruktur og teams, kanal eller chat, etablert
- Systemeier og tjenesteeier etter produksjonssetting er avklart

Prosjektkoordinator har ansvar for:

- Brukerveiledning i TOPdesk (lenke rutinebeskrivelse for knowledge management)
- Driftsdokumentasjon når app skal driftes av ITA legges i OneNote ihht mal der
- Oppdatere kort i systemkatalogen
- ROS-analyse
- RFC

Sjekkliste prosess for utviklingsteamet:

- Avklar oppfølging av fremgang med prosjektkoordinator.
- Avklar realistisk tidsramme, estimert tidsbruk og kostnader med prosjektkoordinator
- Avklar hvem som skal være primær bidragsyter og hvem som skal ha støttende rolle.
- Avklar regler rundt protected main branch, bruk av andre branches, pull requests, review av PRs etc
- Avklar om roller skal byttes underveis.
- Avklar hvor ofte dere møtes for reviews: Ukentlig, daglig eller oftere?

Andre viktige digitaliseringsplattformer:

- TOPdesk, Apache Freemarker
- Rapid Identity
- Drupal, PHP
- Azure
- Microsoft Power Platform, Power Fx

Sjekkliste før prodsetting:

- Se Change-prosess (lenke)



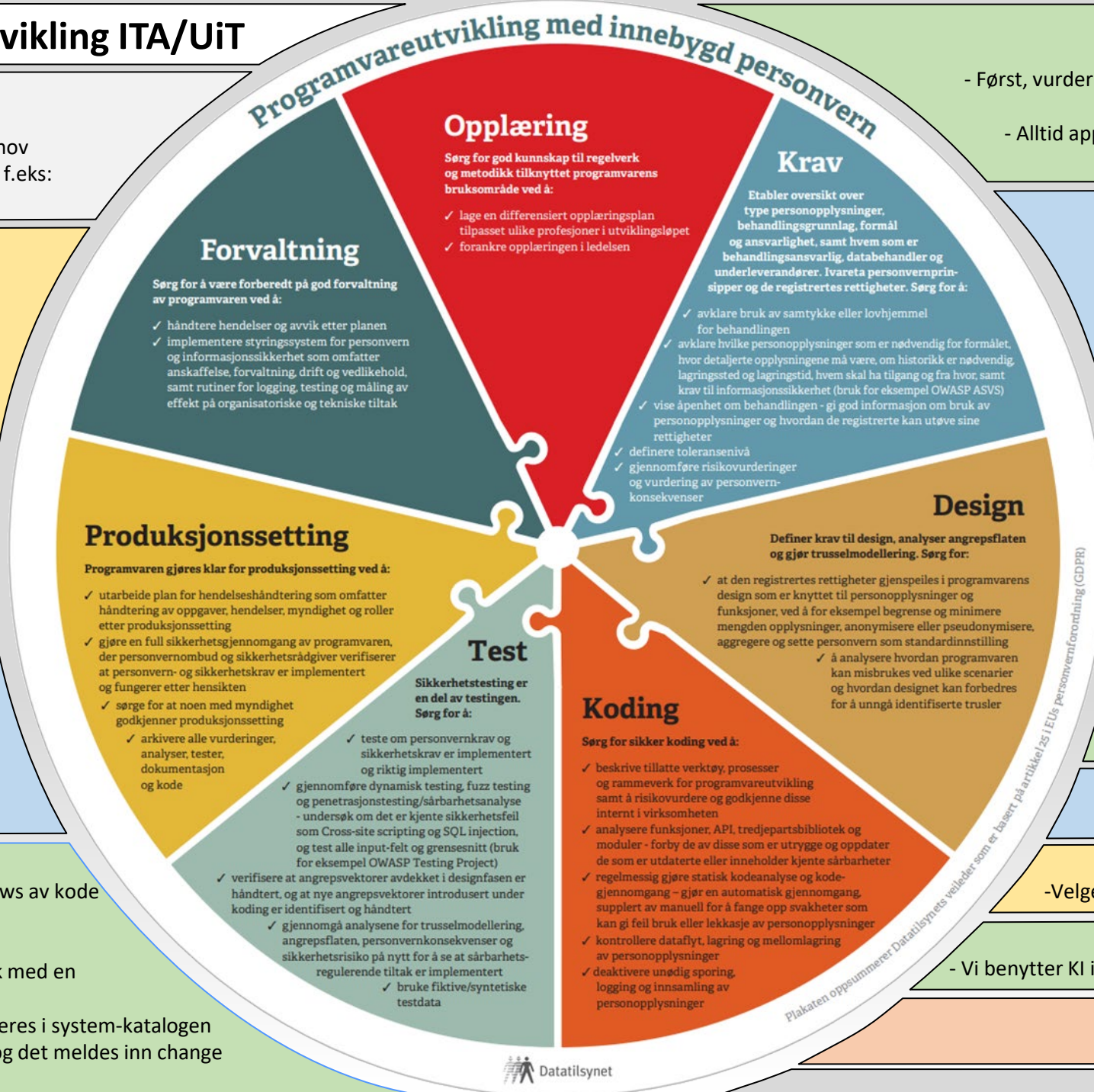
Rammeverk – Systemutvikling ITA/UiT

- Standarder:**
- .NET(C#) er hovedrammeverk/språk
 - Python og JS kan også benyttes ved behov
 - Følg en kodestandard for aktuelt miljø, f.eks:
 - PEP8 kodestandard (Python)

- Dokumentasjon:**
For utviklingsgruppa:
- I egnet format sammen med koden, f.eks. api-dokumentasjon via swagger annotations.
 - Følg praksis ihht: github.com/UiT-ITA/Utviklere-Anbefalt-praksis
- Øvrig:
- Forretningsregler som er implementert
 - Eksponerte grensesnitt/ informasjonsmodell
 - Tilgjengeliggjøring av drifts-dokumentasjon

- Prosess:**
- Antall utviklere ihht behov (2 er foretrukket)
 - Prosjektkoordinator: Teamkoordinator eller tildelt av prosjektet.
 - Kontinuerlig forbedr. av rammeverket.
 - Retrospectives, en gang i semestret.

- Policy:**
- Protected main branch, PRs, peer reviews av kode
 - Grundig dokumentasjon
 - CI/CD der det er hensiktsmessig
 - Arkitekturavsjekk, design review (snakk med en arkitekt)
 - Det som utvikles for produksjon registreres i system-katalogen med ref til tilhørende dokumentasjon og det meldes inn change



- Plattform:**
- Intark for integrasjoner
 - F rst, vurder: PaaS for nyutvikling i sky, deretter Container, til slutt «Lift&Shift».
 - Alltid applikasjonsmonitorering i produksjon, skal v re skybasert (for nyutvikling).

- Utviklingsrammeverk:**
For nyutvikling:
- .NET(C#) er hovedrammeverk/spr k
 - Python(Django, Flask) ved behov
 - Power Automate for RPA
- Annet:** Avklar
- Unng :** NodeJS og de brukt for legacy

- For legacy:**
- Tidligere .NET-baserte rammeverk
 - Java
 - Ruby on Rails
 - Php
 - Etc

- Cloud:**
- Cloud native PaaS hovedretning p  nyutvikling, f.eks. serverless med azure functions. Kan benytte egnede rammeverk som .NET, JamStack eller andre aktuelle (evt. i kombinasjon).

- Repository for kildekode:**
- GitHub

- Digital suverenitet:**
- Velge teknologier ihht behov for digital suverenitet

- KI:**
- Vi benytter KI i de tilfeller og den grad det er hensiktsmessig.

- Unntaksh ndtering:**
- Beskrive hvorfor og avklar aksept.