

DiVA

powered by Cora

2019-11-11



Cora, vad är det?

- Cora är ett namn, som anspelar på det engelska ordet core
- Cora är en teknisk lösning (för system som hanterar metadata med tillhörande binära filer)
- Cora är ett generiskt system, konfigureras för att hantera DiVAs domänområde
- DiVA powered by Cora



Utvecklingsnivåer

- **Cora**
- **DiVA**
 - metadata
 - sammankoppling med classic
- **Integration / Klienter**
- **Användning, batchjobb, datauttag, etc.**



Målsättning med Cora

- **Metadatadrivet system**
 - Enklare hantera de vanligaste förändringarna
 - Uppdaterat maskinläsbart format
- **API:er**
 - Gör det möjligt med olika klienter
 - Integrera i andra system
- **Tydligare arkitektur**
 - Enkelt att byta ut delar
 - Lösa kopplingar till externa beroenden
- **Öppen källkod**
- **Förbättrad kod**
 - Testvänligt
 - Clean code



Metadata

Två olika typer av metadata

Bibliografiskmetadata, är det data som lagras om publikationer, avhandlingar, forskningsdata mm. Lagring, visning, sökning mm i detta metadata är syftet med systemet. I samband med utvecklingen av Cora benämns detta metadata som data (registreringar / poster).

Den andra typen av metadata är den som behandlar vad som kan lagras, vilka typer av registreringar man kan göra, vilka variabler man skall fylla i för respektive registreringstyp, min och max värden, vilka val man kan göra för olika listor, ledtexter, definitioner, mm. Helt enkelt data om systemens data. I samband med utvecklingen av Cora är det denna andra typ av metadata vi menar när vi pratar om metadata.



Delar som behövs för en registrering

För att kunna registrera / visa uppgifter om en post behövs det tre delar

- **Data**
- **Metadata**
- **Presentation**



RecordType

- **Baskomponent för systemet**
- **NameInData**
- **Namn och definition**
- **Pekar ut**
 - Metadata
 - Presentation
 - Sökningar
- **Typ av id**
- **Publikt data**



Data

Data är helt enkelt de uppgifter vi lagrar för en given registrering. Det finns en instans av data för varje registrering i systemet.

Cora har ett generellt dataformat som används för att skicka data in och ut ur systemet CoraDataFormat.



Metadata

Metadata består av två huvudsakliga byggstenar

- **Variabler, som har en typ (collectionVariable, textVariable, etc.) och beskriver en enskild variabel**
- **Variabelgrupper, som är en samling av en till många variabler, eller andra grupper av variabler**

Metadata gör det möjligt att

- **Definiera variabler och dess egenskaper.**
- **Definiera variabelgrupper och därigenom gruppera ihop variabler och andra variabelgrupper.**
- **Definiera metadataTyper**

Både variabler och variabelgrupper kan byggas som hierarkier, där barnet begränsar det som går att göra utifrån föräldern.



Presentation

Presentation består av några huvudsakliga byggstenar

- **Text**
- **Gruppering (container)**
- **Referensvariabel**
- **Referensgrupp**

Presentationen gör det möjligt att

- **Bestämna ordning på variabler och variabelgrupper ur den grupp som presenteras.**
- **Lägga till ledtexter som rubriker mm.**
- **Utöka presentationen av metadata med ytterligare grupperingar (containers)**
- **Högnivå styling, rubriker, kursiv, bredd, m.m.**



Metadata – andra metadata

Förutom variabelmetadata finns det också metadata som beskriver

- **Vilka sökningar som går att göra**
- **Vad som indexeras**
- **Registreringstyper**
- **Funktioner (indexering)**
- **m.m.**



MetadataGroup

- **NameInData**
- **Namn och definition**
- **Barn**
 - Referens till barnet
 - Min / max
 - Indexering
 - Rättigheter



Variabler

- **NameInData**
- **Namn och definition**
- **Möjliga värden (lista eller regexp.)**



ActionLinks

- **Read**
- **Update**
- **Delete**
- **Index**
- **IncomingLinks**

- **Create**
- **Validate**
- **List**
- **Search**



API förutsättningar

- **REST-API**
 - Stateless
- **Tre nivåer av format**
 - JSON, (XML kommande utökning)
 - Cora format
 - DiVA specifikation
- **Erbjuda presentationer**
 - Ej tvingande användning
- **Förändringar i DiVA specifikation utan kodförändringar**
 - Lösningar där användare matar in data, bör läsa specifikation



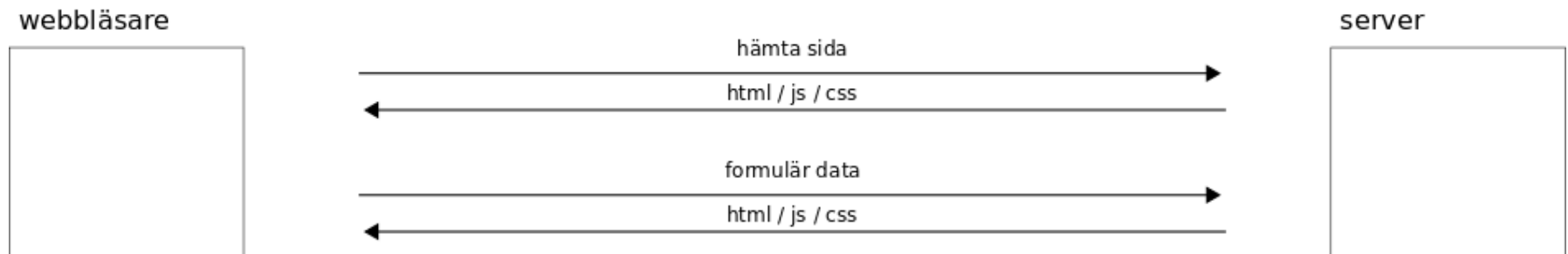
REST-API

- **Hela systemet är åtkomligt via API:et**
 - Definitioner för allt (posttyper, listor, texter, etc)
 - Data (med actions)
 - Funktionsdefinitioner (sök)
- **Målsättning att fullt ut kunna stödja andra klienter**
 - Andra plattformar (Android, ios, etc)
 - Integration mot andra system
 - Servrar som klienter
 - Andra som vill bygga egna gränssnitt
- **Möjlighet att förändra formulär och presentation utan förändring i klienter**

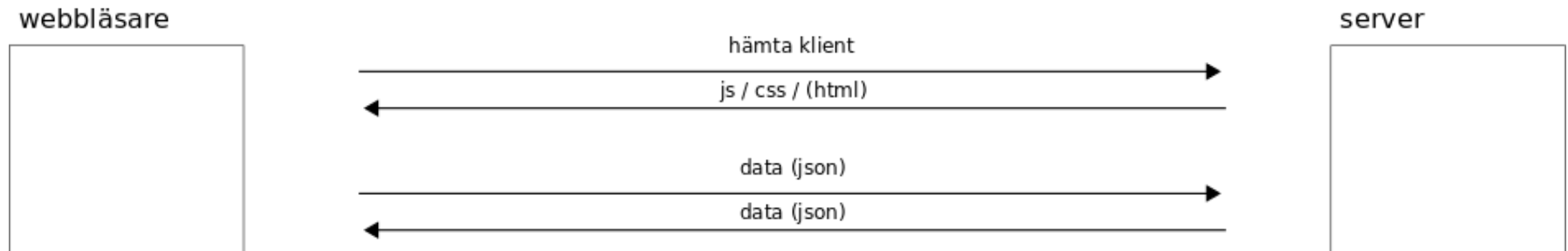


Cora har en klient

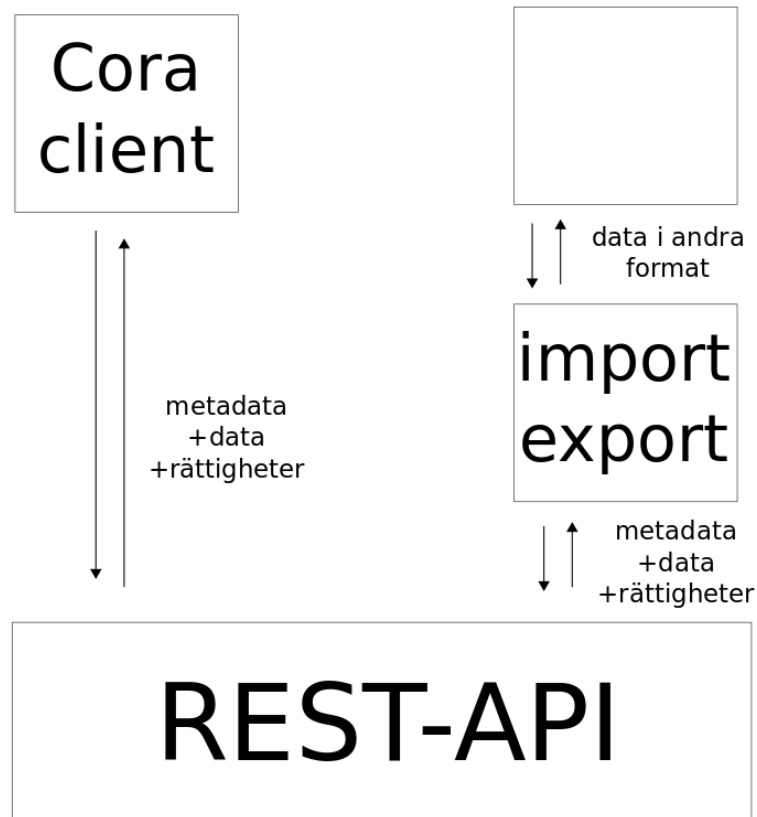
Nuvarande DiVA



DiVA powered by Cora

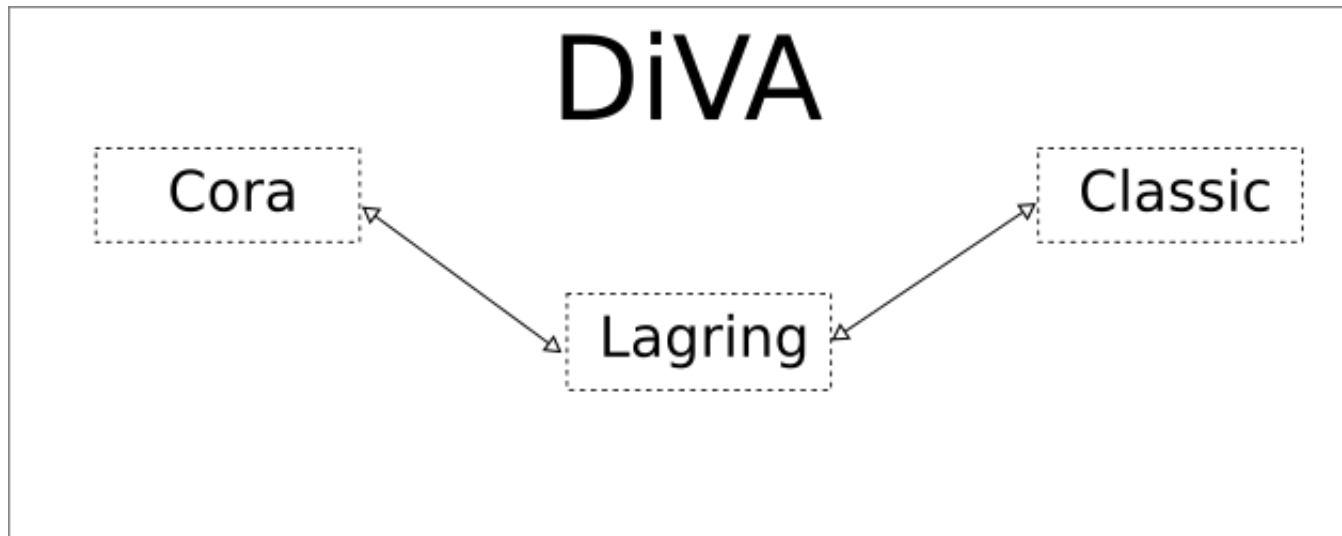


Cora API vs Import Export



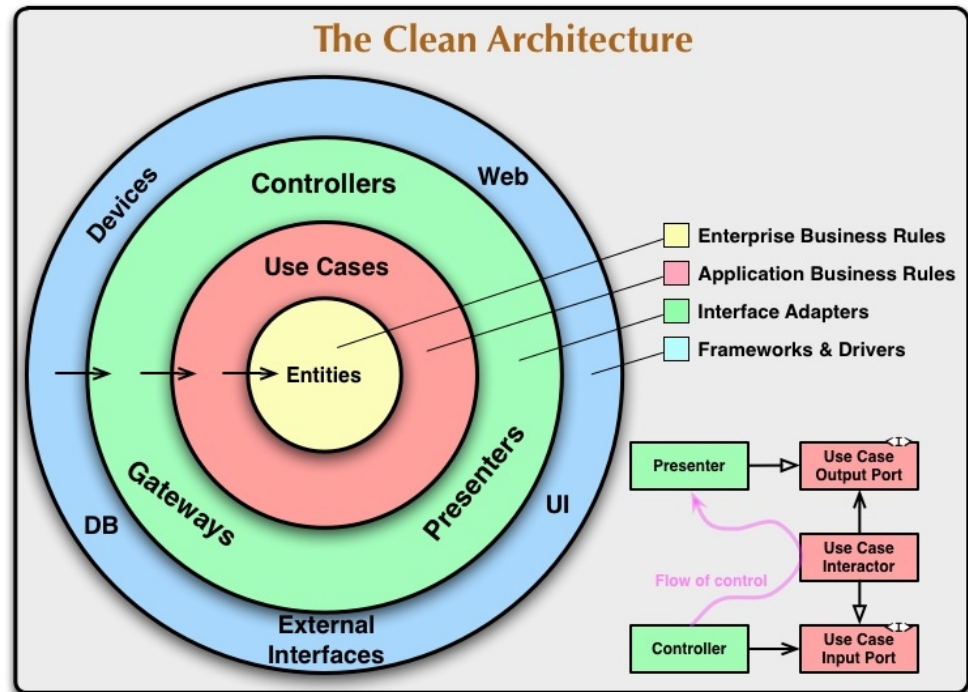
Två tekniska lösningar, ett system

- DiVA “Classic”
- DiVA “powered by Cora”



Arkitektur - The Clean Architecture

- Independent of Frameworks. The architecture does not depend on the existence of some library of feature laden software. This allows you to use such frameworks as tools, rather than having to cram your system into their limited constraints.
- Testable. The business rules can be tested without the UI, Database, Web Server, or any other external element.
- Independent of UI. The UI can change easily, without changing the rest of the system. A Web UI could be replaced with a console UI, for example, without changing the business rules.
- Independent of Database. You can swap out Oracle or SQL Server, for Mongo, BigTable, CouchDB, or something else. Your business rules are not bound to the database.
- Independent of any external agency. In fact your business rules simply don't know anything at all about the outside world.

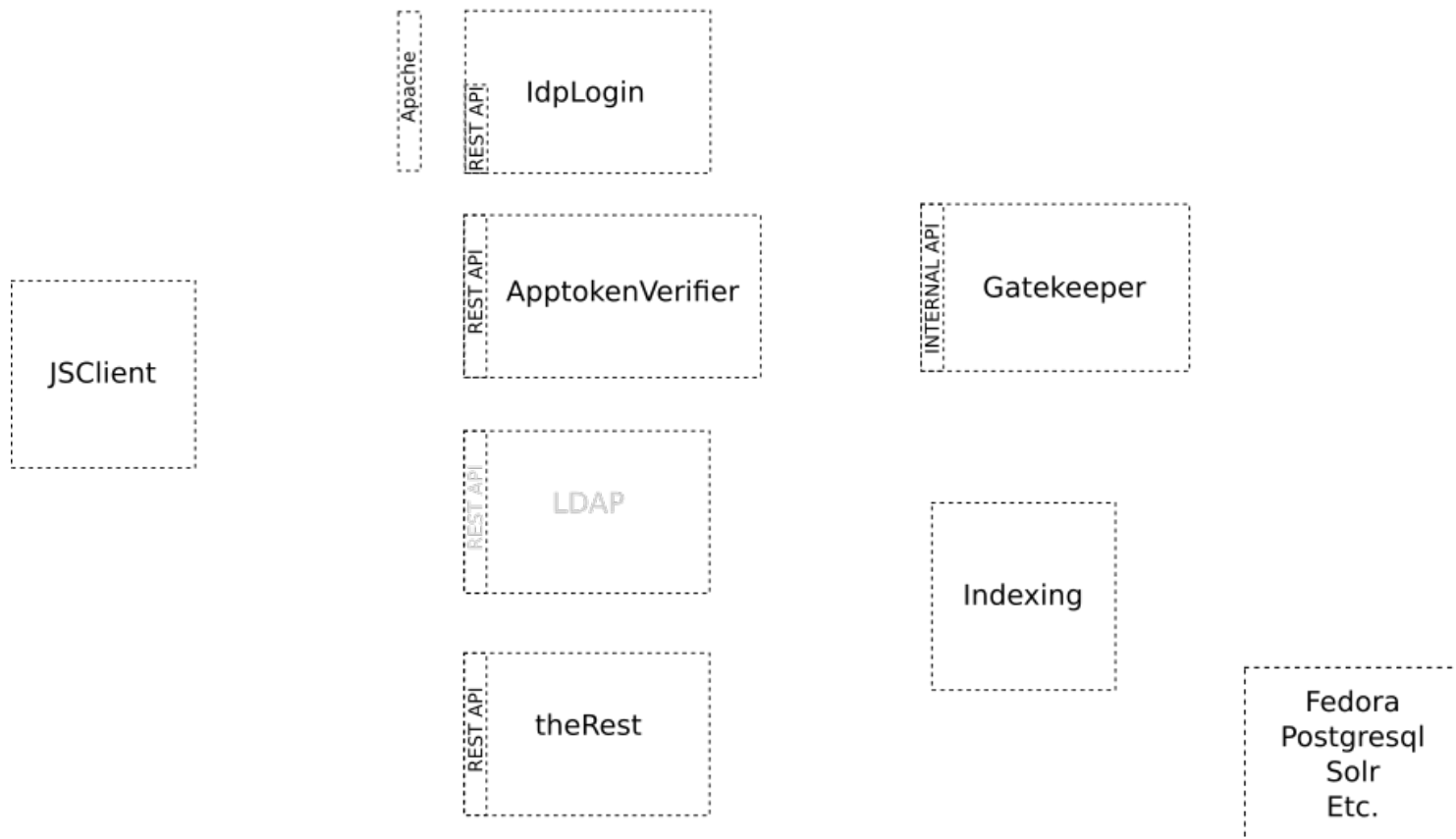


Robert Martin - <http://blog.8thlight.com/uncle-bob/2012/08/13/the-clean-architecture.html>



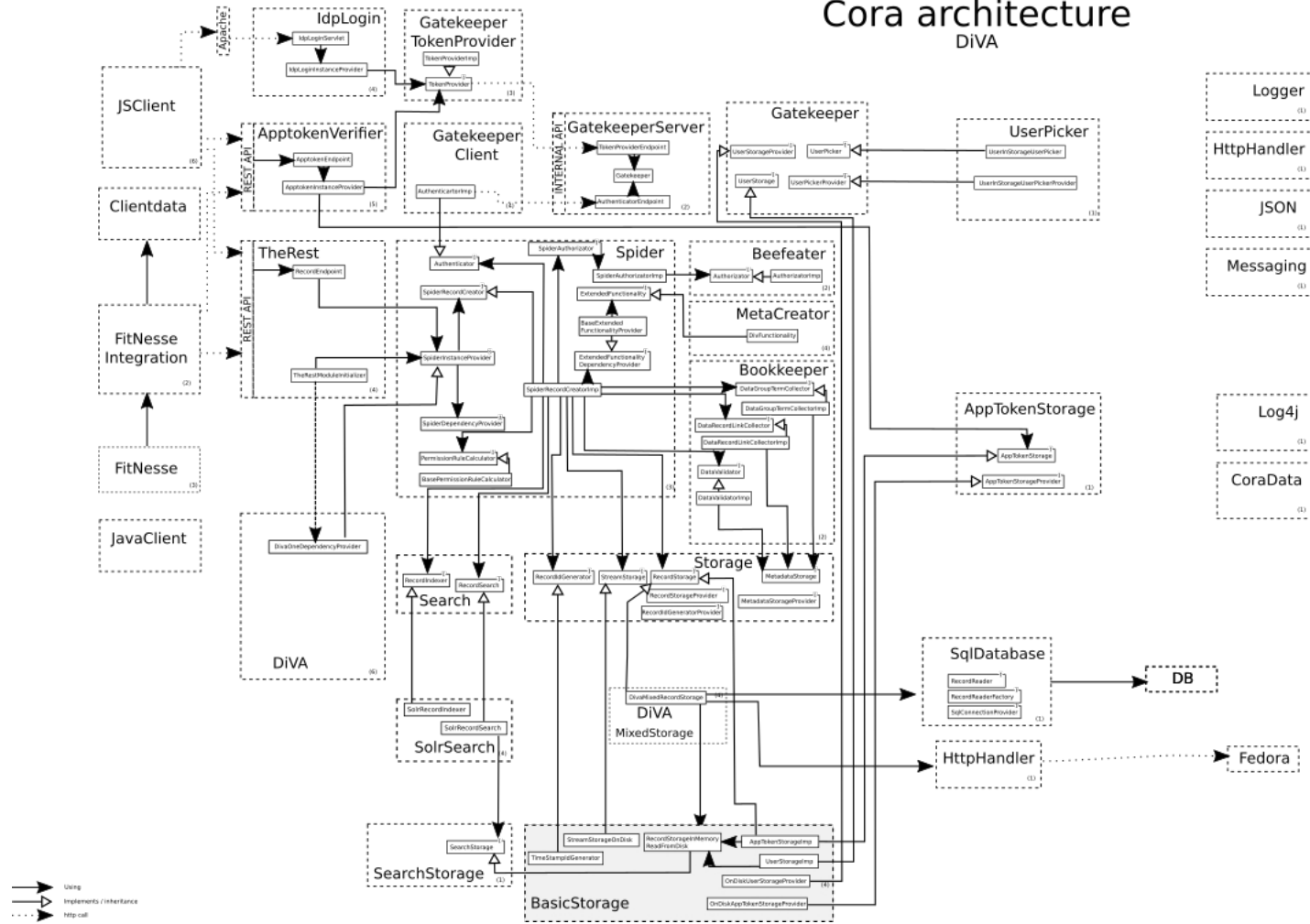
Driftsättning

Cora deployment



Komponenten

Cora architecture DIVA



Öppen källkod Cora

- **GPL v3 (eller senare)**
- **Publikt på github**
 - <https://github.com/lisu-ub-uu>
- **Vad är utlagt?**
 - Kod
 - Dokumentation
 - Acceptanstester
 - Ärendehantering



Agil systemutveckling

- **Ökad transparens för utvecklingsprocessen**
- **Baserat på XP och Scrum**
- **Inkrementell release efter varje sprint**
 - Inte nödvändigtvis kompletta funktioner
 - Det som är med är klart för produktion (done)
- **Vad som utvecklas näst, bestäms i samarbetete mellan beställare och utvecklare**



Förändringar i utveckling

- **TDD**
- **100% code coverage med unit tester**
- **Clean architecture**
- **Clean code**
- **0 fel / varningar i SonarQube (vår profil)**
- **Parprogrammering**



Olika miljöer

- **Development**
- **Build**
- **Preview**
- **“Test”**
- **Produktion**



Prioritetsordning utvecklingen

- **Bygg fungerande och testade delar, men med begränsad funktionalitet**
- **Prioritera utifrån högsta värde för systemet som helhet**
- **Se över prioriteringen i samband med varje sprints slut / sprintstart**



DiVA powered by Cora, status

- Definitionerna i Cora byggs så att de matchar nuvarande innehåll
- Parallella system i test, läser(skriver) samma data
- Börja med att kunna visa organisation-, personposter i DiVA powered by Cora
- Bit för bit
- Lugnt och metodiskt
- Anpassa eftersom



Länkar

- **Test klient:** <https://cora.test.diva-portal.org/diva/jsclient/>
- **Test server:** <https://cora.test.diva-portal.org/diva/rest/>

- **Preview klient:** <https://cora.epc.ub.uu.se/diva/jsclient/>
- **Preview server:** <https://cora.epc.ub.uu.se/diva/rest/>
- **Fitnessse:** <https://cora.epc.ub.uu.se/diva/fitnessse/TheDiVATests>
- **Jira:** <http://jira.epc.ub.uu.se>
- **Github:** <https://github.com/lisu-ub-uu>



