



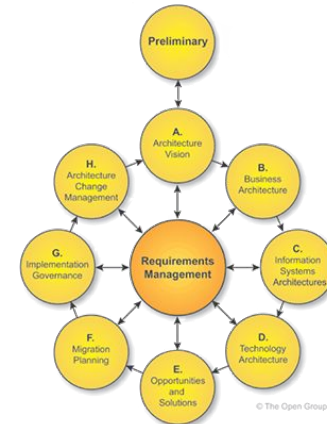
Beneficiary Enrollment migration

May 27th 2020



Project global approach

- With a fix scope, fix budget and fixed timeline, we can't claim we are agile anymore (scope was not 100% fixed in previous migration phase: '2 modules migrated')
- ... but we will keep the same (togaf-based) iterative approach
- ... 5 iterations:
 - Dev readiness
 - Persons & Families
 - Policies
 - Contributions
 - Wrap up

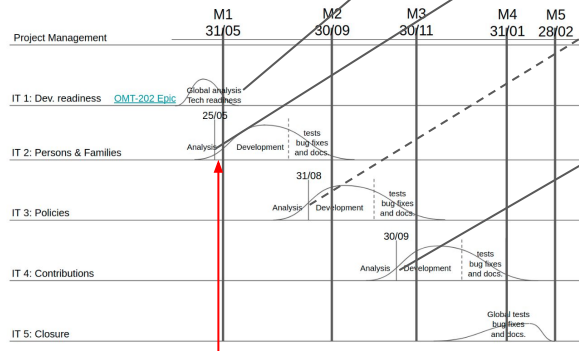


Beneficiary Enrollment

Slide from Gumzo 04/05:

Notice D. (RFA-2019016): Beneficiary enrollment

- **Scope:** migration 'as is' of persons & families, policies and contributions
- **Timeframe:** ~~12~~ 9 months (mid May 2020 - mid Feb 2021)
- **Methodology:** continuation of claim and locations & HF's migration



<https://openimis.atlassian.net/wiki/spaces/OP/pages/1365934087/Beneficiary+Enrollment>

<https://openimis.atlassian.net/wiki/spaces/OP/pages/1366228996/Persons+Families+module+scope>

<https://openimis.atlassian.net/wiki/spaces/OP/pages/1365999627/Policies+module+scope>

<https://openimis.atlassian.net/wiki/spaces/OP/pages/1366163471/Contributions+module+scope>

we are here... **but contract is still not signed!!**

Module migration approach

- Scope (wiki) and prepare screen mocks (sandboxes)
 - delegate users' feedback gathering to implementers (presentation made on 27/05)
 - ensure the scope identified all features (screens, stored procs, batches, ...) : 'between us' review (right now)
- "Let's do it" (2-3 sprints of 2 weeks):
 - Coding (+ unit tests) in dedicated [github](#) branches - following [agreed Gitflow](#) (with *-rc versions)
 - Prepare tests (data sets, ...) - contribute to [QA wiki](#)
 - Deploy on release.openimis.org
- Tests & documentation phase
 - On release.openimis.org, execute test plan and fix accordingly
 - Provide module (technical) documentation, update user guides,...

About modules

- **Backend modules**

- **DON'T TOUCH DATABASE STRUCTURE** (nearly)
- (django) model extended: security, search capabilities, configuration options,...
- GraphQL
 - additional query (search) capabilities
 - implement mutations (expose django signals) & service layer

- **Frontend modules**

- Add 'Published components' (pickers,...)
- Add 'Contribution points' (and implement ref. Contributions from other modules)
- Add redux events (actions/reducers) and states
- Replaced proxy screens

Thank you !



www.bluesquarehub.com

info@bluesquarehub.com