

accelerate IA Cloud Enterprise / Use Case delivery with

Open Development Framework



Open Development Framework

1. Problem statement >
2. Definition
3. Architecture
4. SDLC
5. Reusable components
6. Adoption and contribution

Delivery takes long time and high-skilled people

Challenging for Business



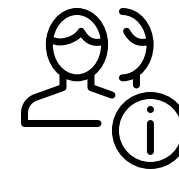
Tech drives Business Process design

Platform concepts outdated and push for complicated BP design (Split/Merge, integration sub-processes, etc.). BP «unreadable» for business



Lack high-level OOTB components

Team takes time to build library of components, only then use case itself



Manual release cycle generates errors

Release management is manual, slow and varies for BP parts



Large projects are hard and expensive to manage

Challenging for Delivery Teams



Human factor

Syntax is not a standard.
Majority of developers
doesn't like XML/Groovy
mix



Productivity

Extra efforts while crafting
bots: manipulate data flow,
expose vars for next step,
etc.

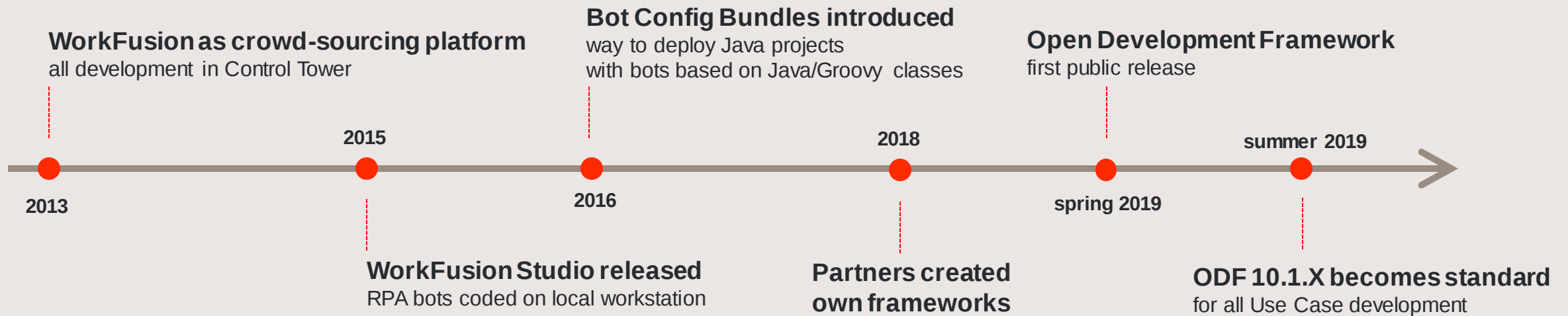


Code support

Can find 10K-line long
ugly bots across
implementations



Journey to Framework



Open Development Framework

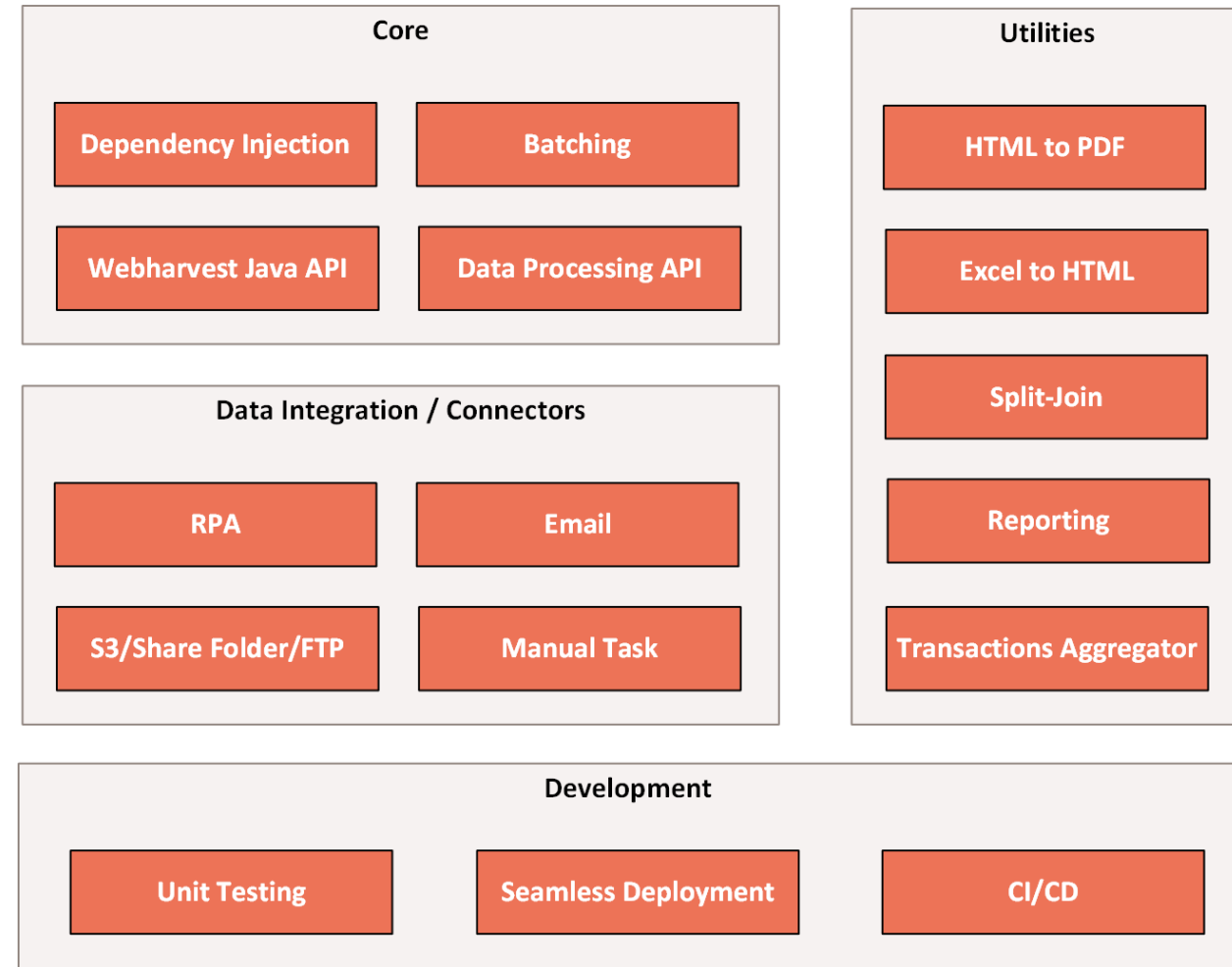
1. Problem statement
2. Definition >
3. Architecture
4. SDLC
5. Reusable components
6. Adoption and contribution

Definition

Open Development Framework is a powerful use case automation development framework used for WorkFusion IA Cloud Enterprise and Use Cases.

Design Principles:

- Reusability
- Extendibility
- Good at Scale
- Designed for RPA Engineers



Benefits

Less implementation effort

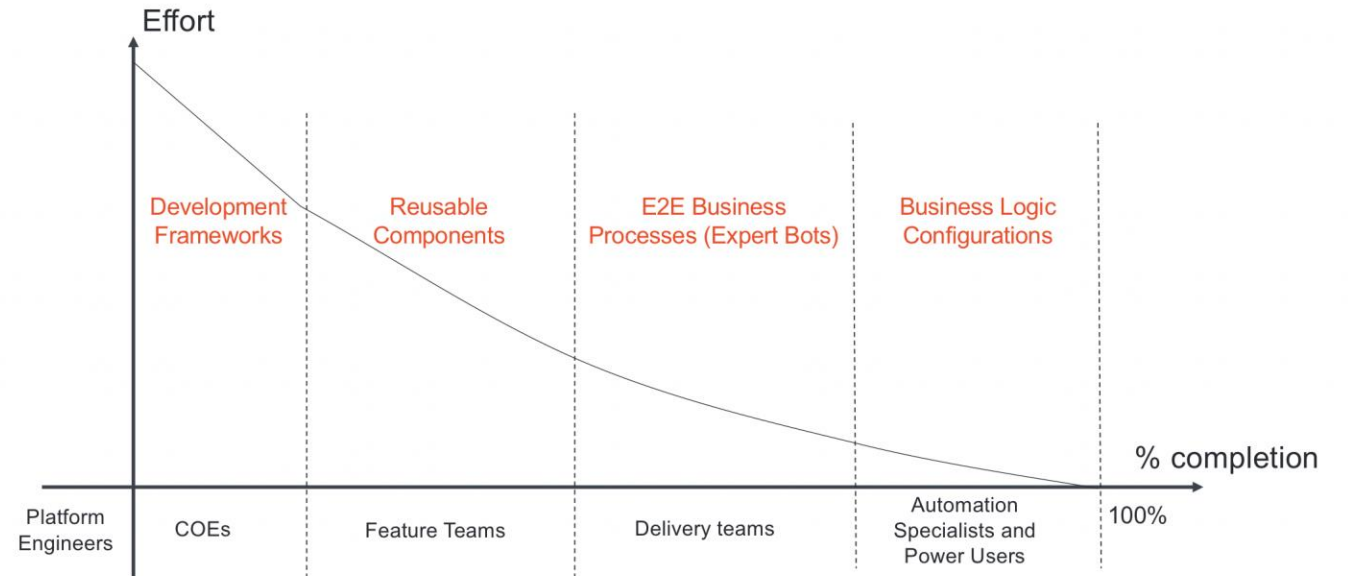
- 20%-50% development time reduction
- Standard library of components

Less support and maintenance

- Seamless way to publish changes
- CI/CD
- Integration Testing for Business Process

Aligned with Product Roadmap

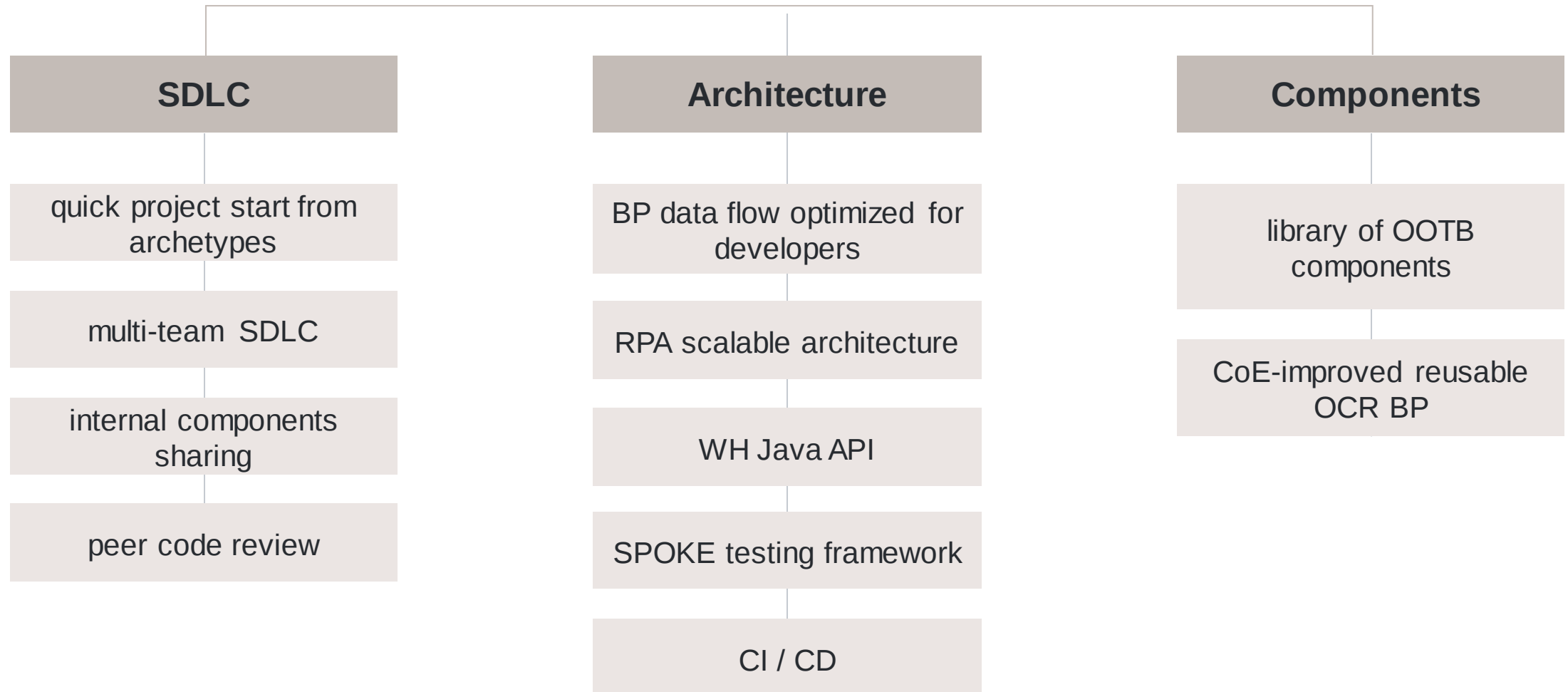
- Becomes part of standard Product shipment
- Backward compatible



Open Development Framework

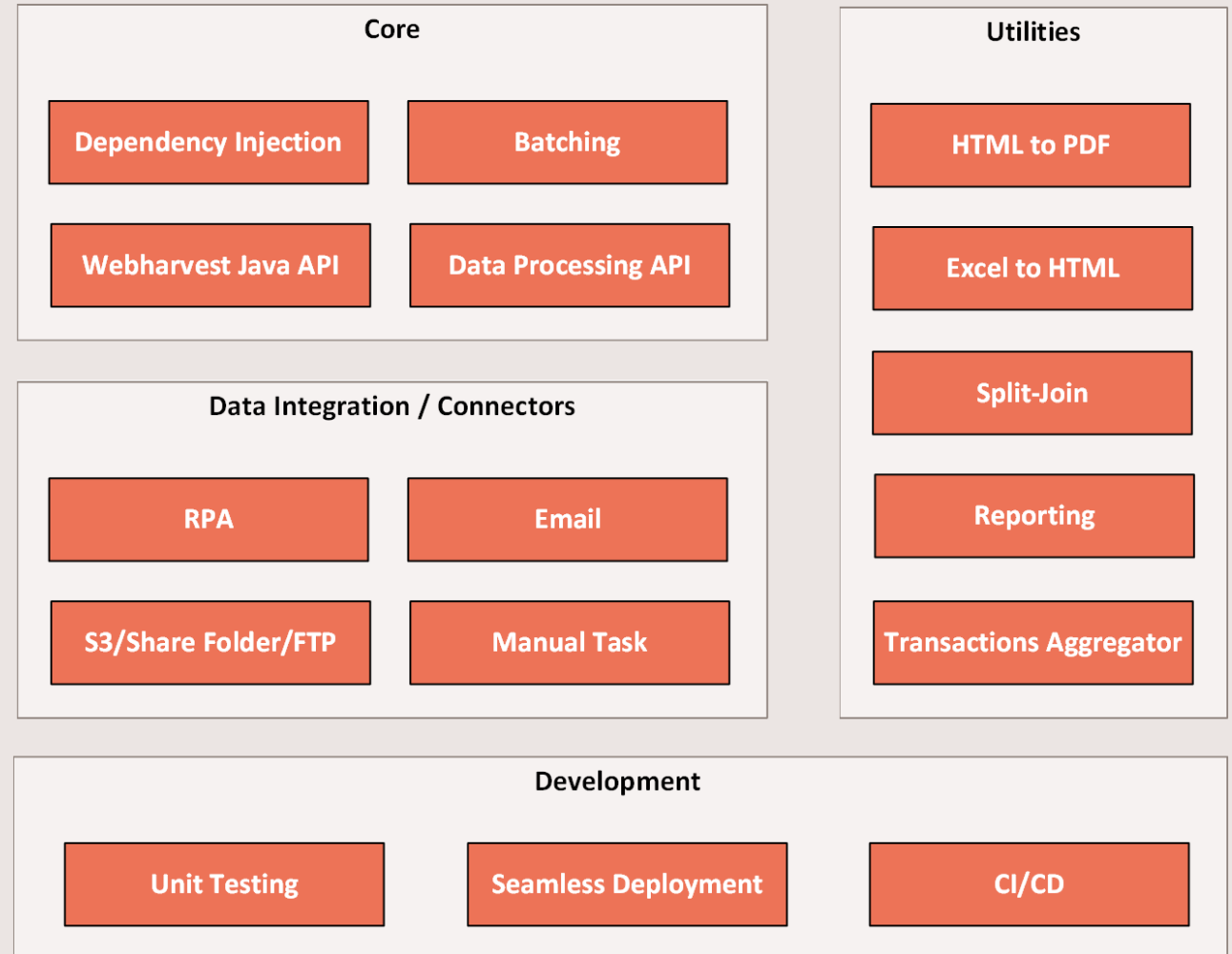
1. Problem statement
2. Definition
3. Architecture >
4. SDLC
5. Reusable components
6. Adoption and contribution

Framework: 3 pillars

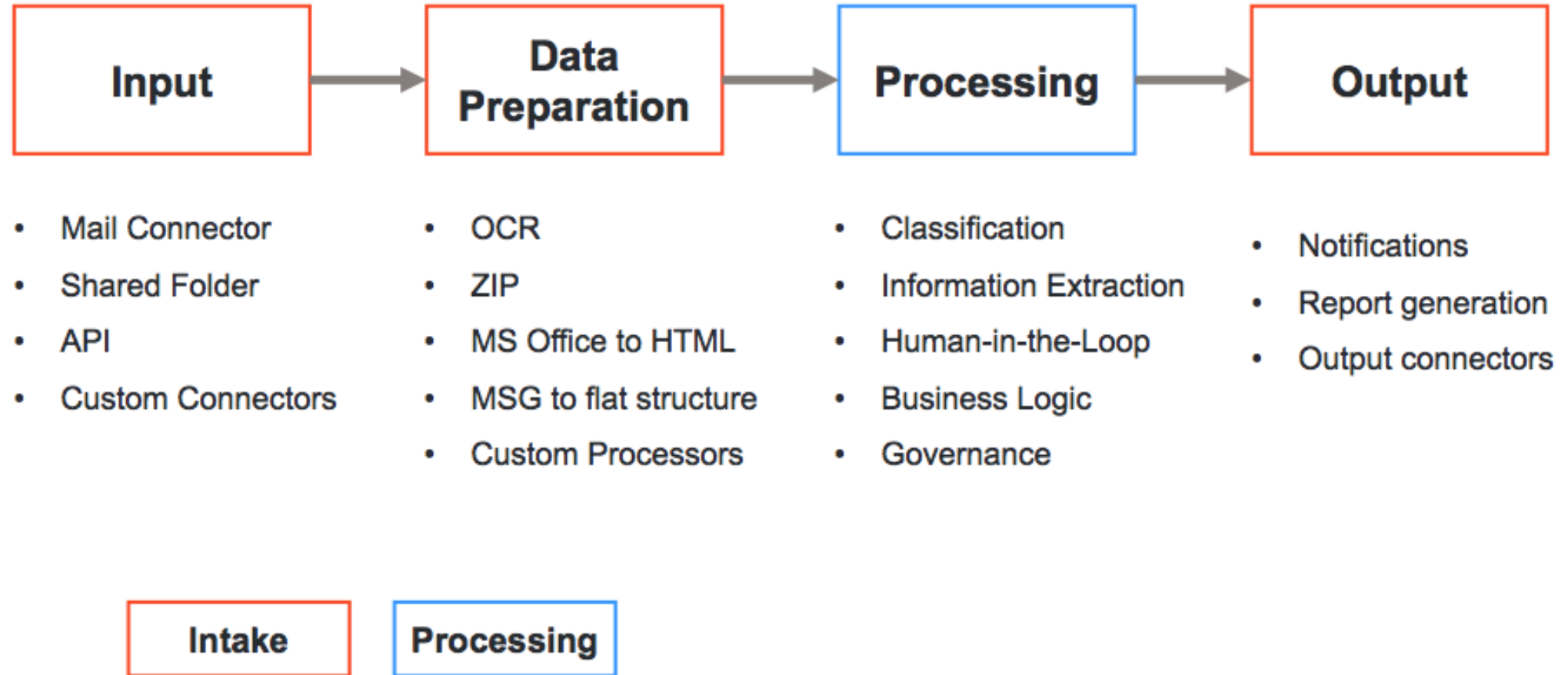


Framework architecture

- Clear **standardized** Java project structure, naming, versioning
- **Reusability** and **collaboration** tools are in Framework DNA
- **Java** is main bot coding programming language
- **Dependency injection** and **extendibility** design principles
- Unit and BP integration automated **testing available**
- Automated **deployments**



ODF in generic business process flow

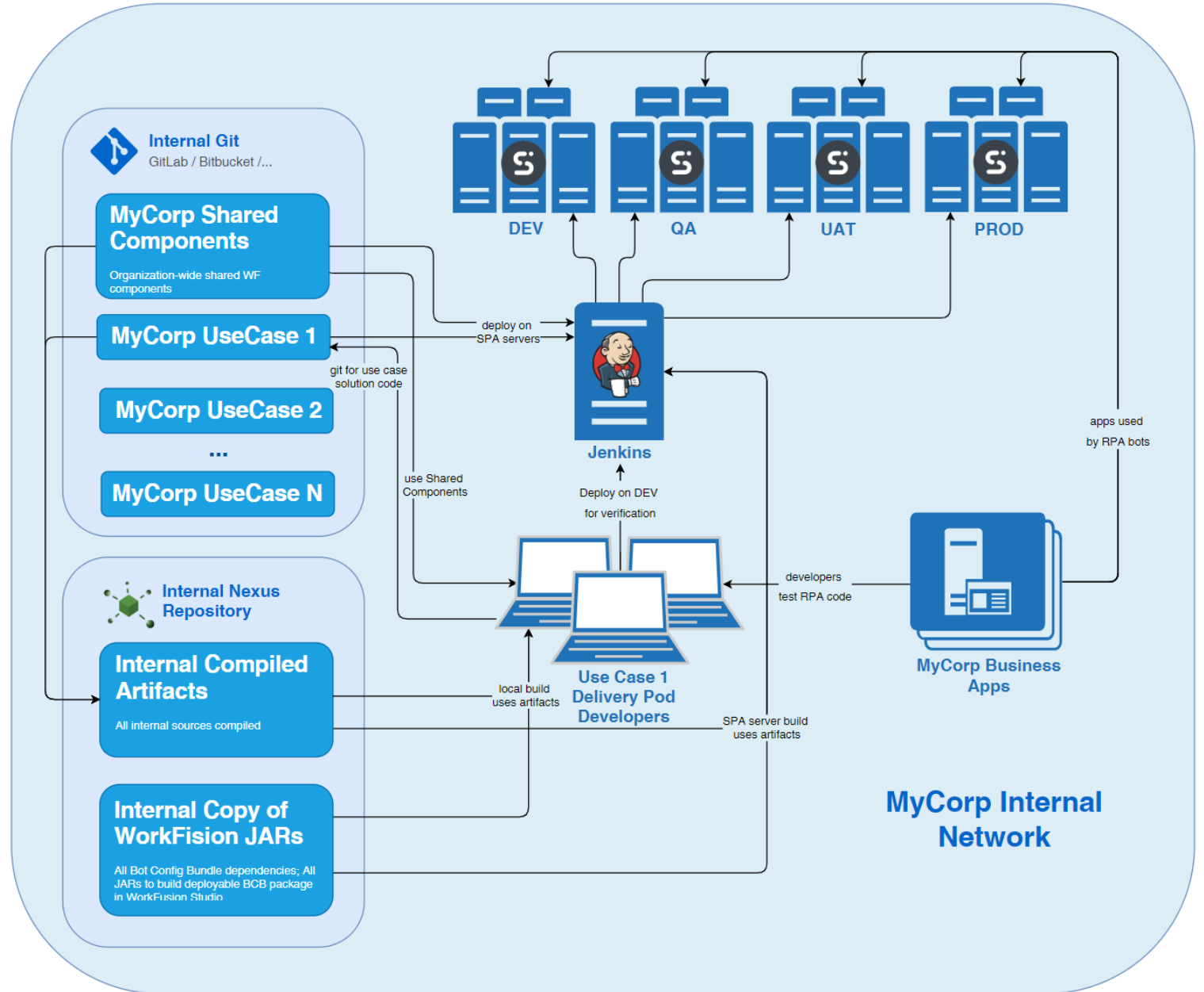


Open Development Framework

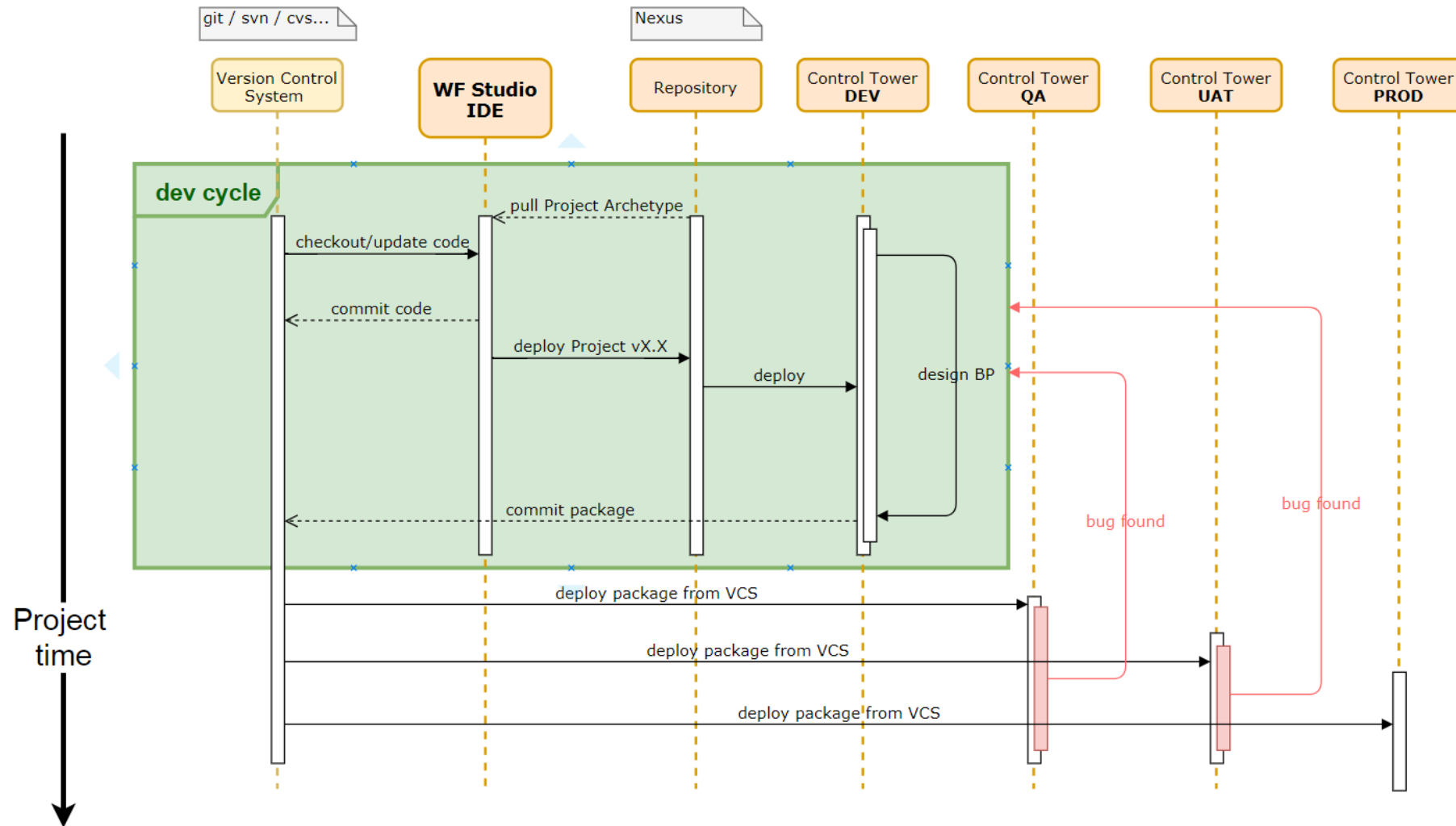
1. Problem statement
2. Definition
3. Architecture
4. SDLC 
5. Reusable components
6. Adoption and contribution

Delivery pod SDLC

- Release build from tagged code
- Auto-deploy build from git
- Reusable components shared as JARs
- Avoid any changes in Control Tower
- Any change/bugfix requires re-deploy
- Pod have access only to required repos



Implementation lifecycle



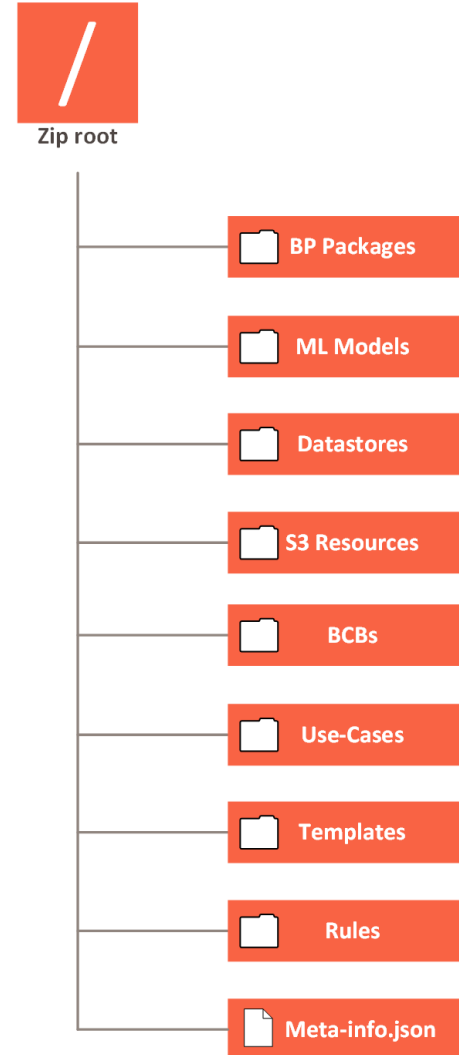
Deploy in single bundle

Bundled resources:

1. Business Process
2. ML models
3. DataStores
4. S3 Resources
5. Bot Config Bundles
6. Other resources (use-cases, rules, templates)

Different conflict strategies supported:

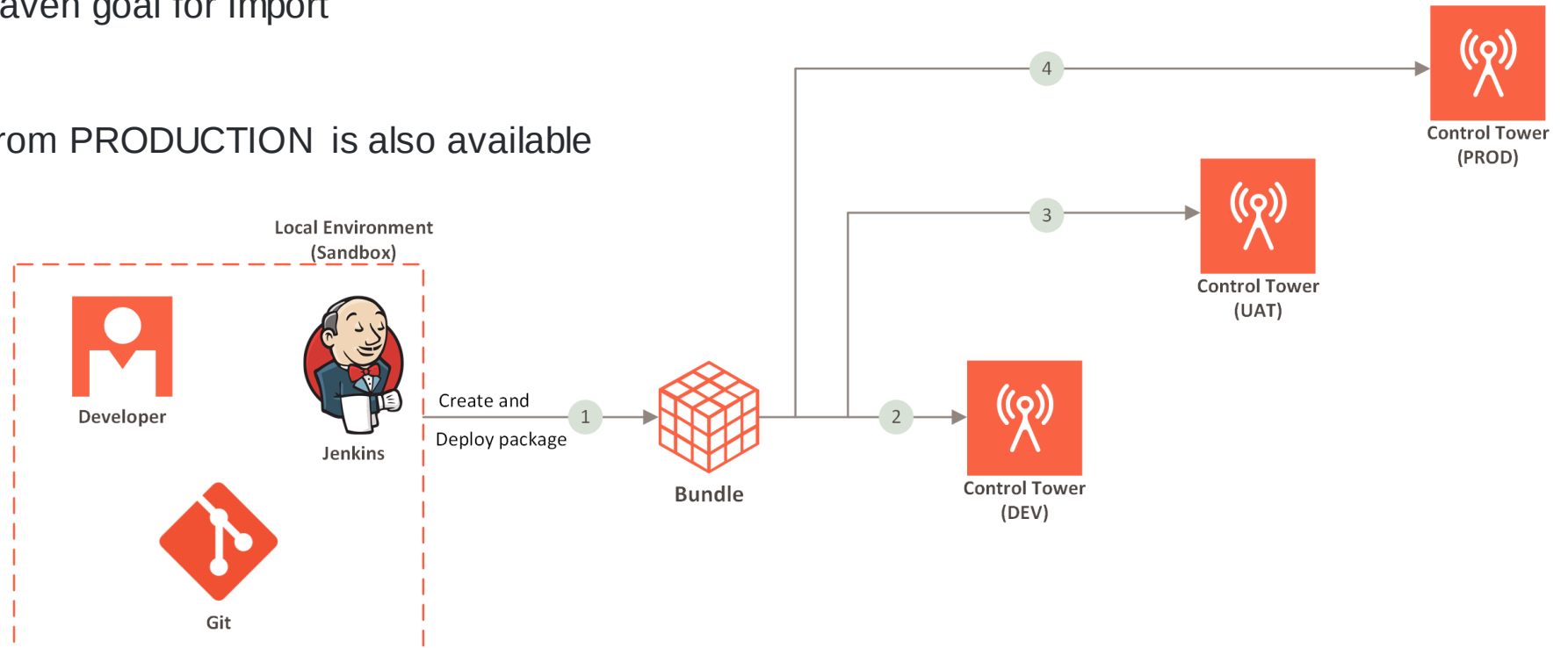
1. Replace
2. Merge (DataStore)
3. Skip



Automated deployment kinematics

1. Create and package bundle (bundle can include full use-case or incremental change)
2. Promote bundle to DEV environment using REST API
3. After development is completed, move bundle to UAT
4. If UAT is passed, promote to Production
5. Developers can use Maven goal for Import

API for Export ML model from PRODUCTION is also available



Open Development Framework

1. Problem statement
2. Definition
3. Architecture
4. SDLC
5. Reusable components 
6. Adoption and contribution

REUSABLE components



- **Connectors** (Share Folder, Email)
- **Converters** (Excel to HTML, HTML to PDF)
- **Processors** (Zip To Files, EML files)
- **Transactions** (Wait, Delete)
- **Documents Split-Join**
- **OCR process** (improved TOD)



Library expected to grow with
WorkFusion and Partners contributions



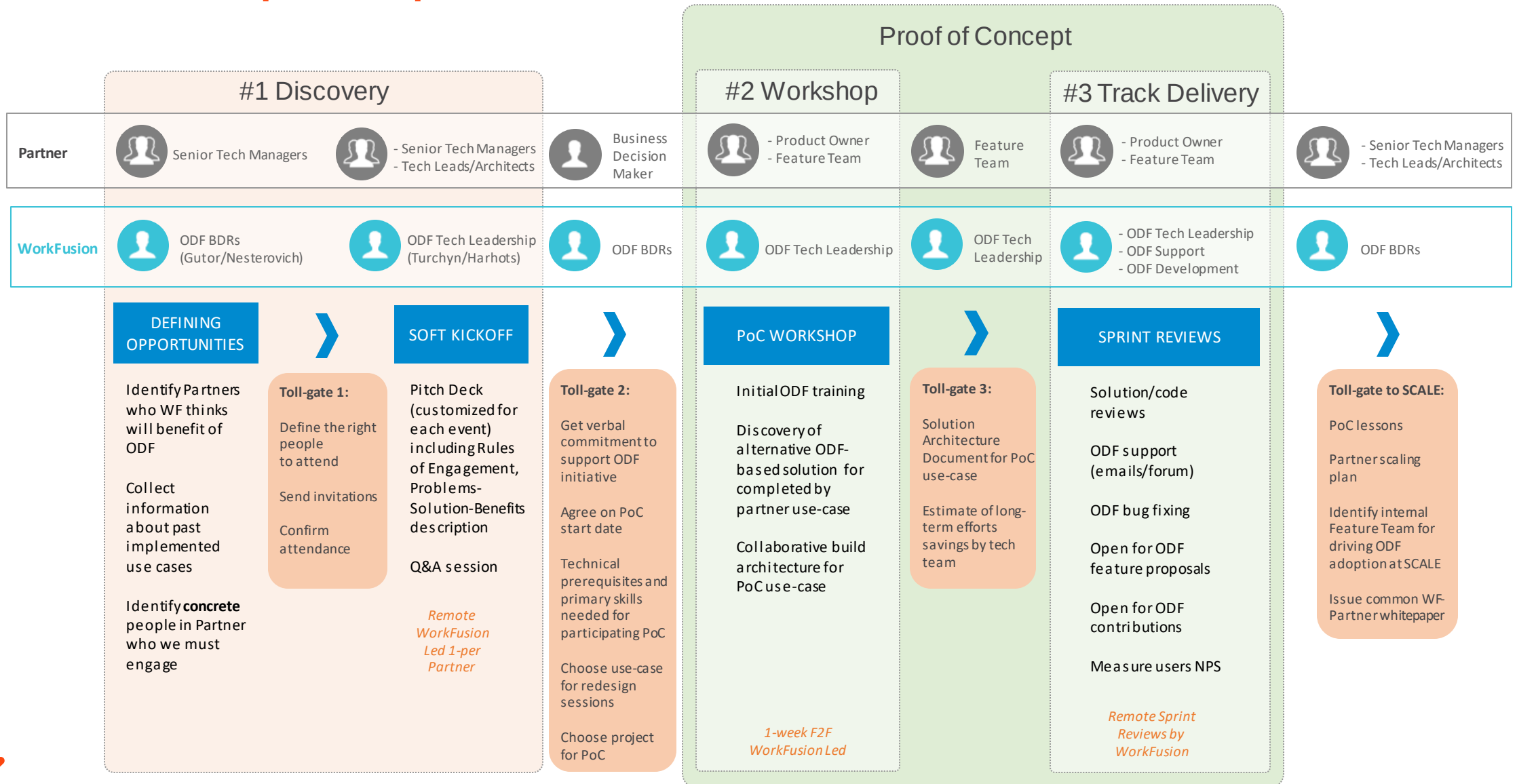
Quickstart Archetype
RPA project Archetype
Tutorial ODF usage automation project



Open Development Framework

1. Problem statement
2. Definition
3. Architecture
4. SDLC
5. Reusable components
6. Adoption and contribution

Partner adoption plan



Contribution

Open Source

- Framework source code is available at ODF documentation
- Pull request workflow for contributions

Library of components

- Components from real implementations can be contributed
- Well documented
- Publicly available



Thank you

Questions?



Andrei Harhots
Director, Product Management
aharhots@workfusion.com