

End User Perspective

David White

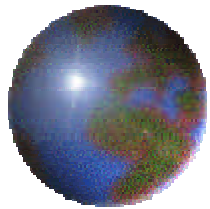
Technology Manager, webMethods

Johnson & Johnson

Ajay Anand

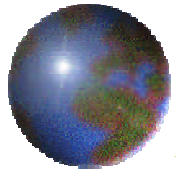
Manager - Architecture & Integration Services

Johnson & Johnson



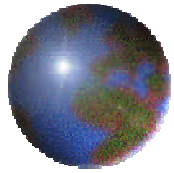
Building an Enterprise Integration Strategy

David White

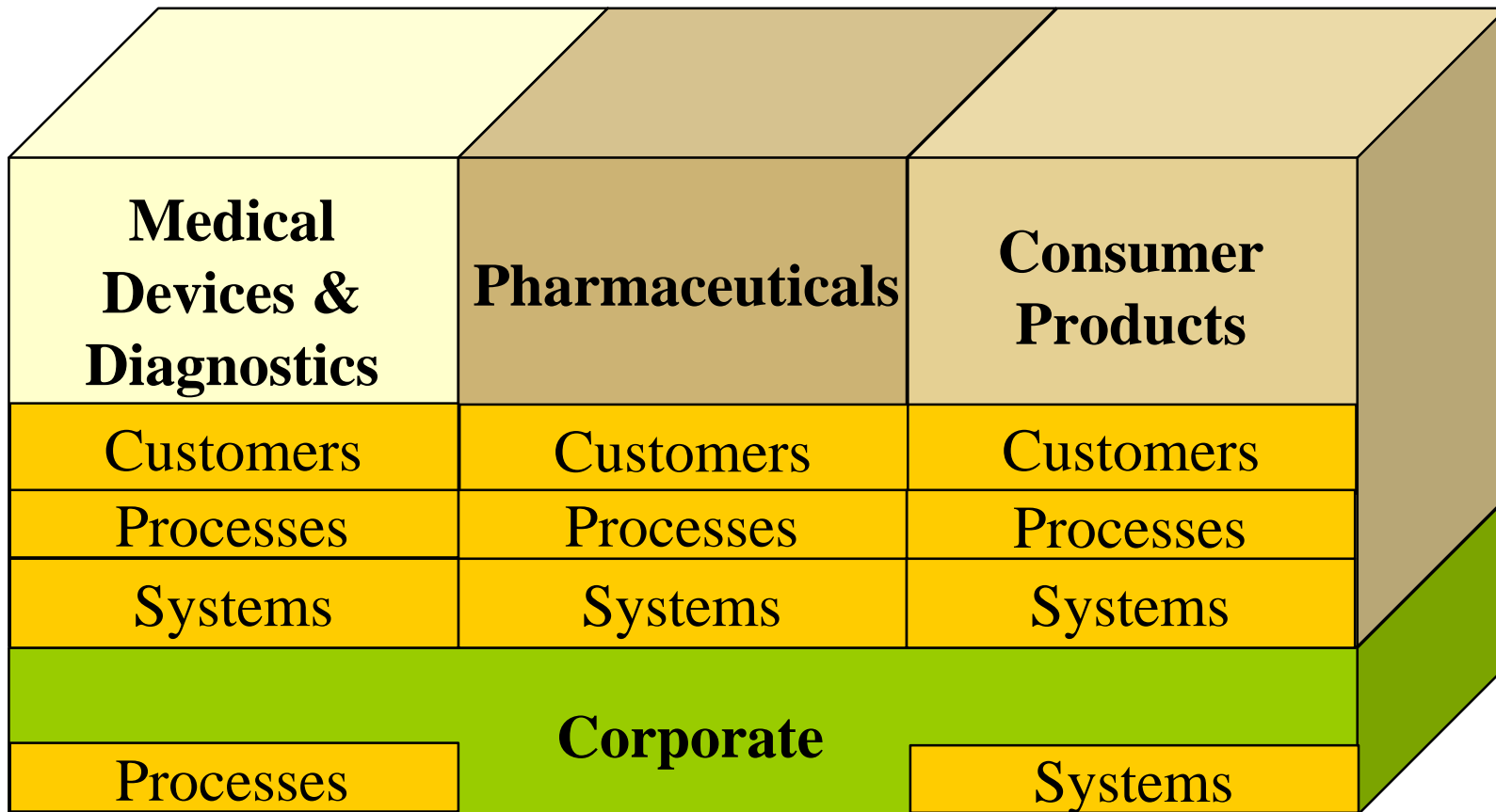


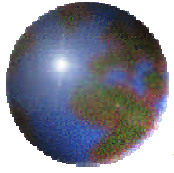
Johnson & Johnson

- Diversified healthcare company founded in 1886 in New Brunswick, New Jersey.
- More than 200 operating companies in 54 countries.
 - International expansion started in 1919 with Johnson & Johnson Canada
 - Companies established in Latin America, Europe, Africa and Australia for more than 50 years.
- Company was family-owned until listed on NYSE in 1944.

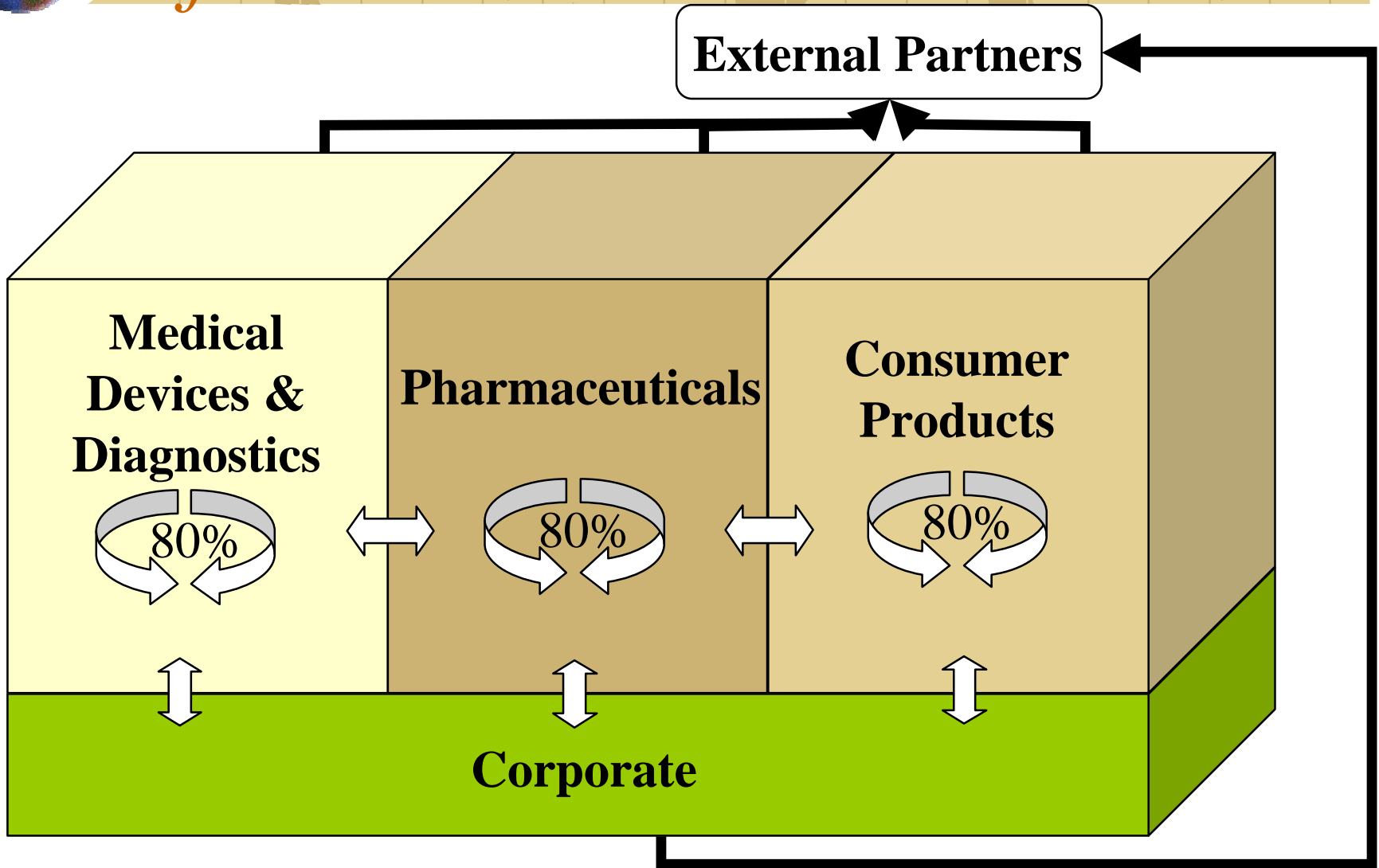


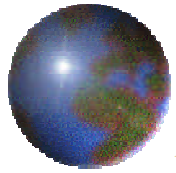
Independent Businesses & Systems





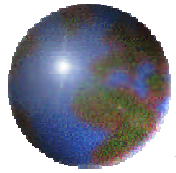
Information Flow





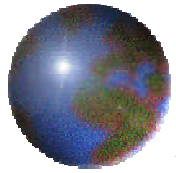
Funding

- Project oriented funding model.
- Contrary to almost all other systems in JNJ, webMethods' infrastructure was deployed as a centralized shared service with cost recovery model.
 - ❑ No overall fee paid by every company to fund infrastructure.
 - ❑ Must prove value to the enterprise on a project-by-project basis.
- Decentralized IM.
 - ❑ No centralized development organization.
 - ❑ Projects must develop code themselves.



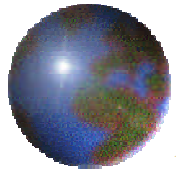
Early Experiences

- Decentralized development resulted in a plethora of:
 - Methodologies
 - Project Plans
 - Documentation Standards
 - Naming Standards
 - Coding Standards/Organization
 - Error handling / Reporting facilities
 - Little reuse



Total Business Integration

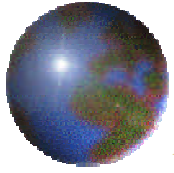
- The challenges.
 - ❖ How can we design integration today that will maximize reusability of data for the integrations of tomorrow?
 - ❖ How can we design integration today that will minimize the negative effects of changing or adding systems in the future?
 - ❖ How can we reduce current project design and development costs?



Total Business Integration

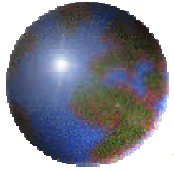
■ The Solution.

- Create a process-oriented integration framework that is “future-proof” and seamlessly links our heterogeneous business applications to facilitate the sharing of information internally and externally including partners, customers and other stakeholders.



Assumptions

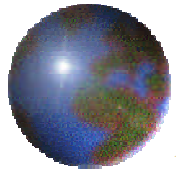
- ✦ An integration can only be properly understood in the context of a business process.
- ✦ Standardizing messages is the key to maximizing reusability while at the same time minimizing the negative impact of changing or adding systems to an integration.
- ✦ Adopting a standard message structure that has the support of a large number of software companies provides the most flexibility, acceptability, and durability.



Value Proposition

■ Reusable architecture and processes

- ❏ Reduced integration time & costs for initial and follow-on projects.
- ❏ Standard methodology and resulting documentation stored in a repository maximizes leveraging.
 - Especially valuable in decentralized development!
- ❏ Common vocabulary facilitates knowledge transfer across the enterprise.
- ❏ Setting the standard for future integration.



Value Proposition

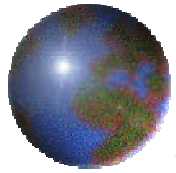
- **Reduced complexity**

- Minimizes point-to-point interfaces.

- Long term reduction in change management and maintenance costs.

- **Potential buffer for affiliates from future changes in application architecture.**

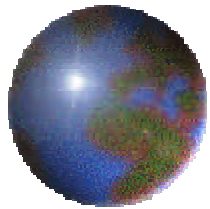
- **Maximizes our middleware investment and instantiates the use of XML.**



Value Proposition

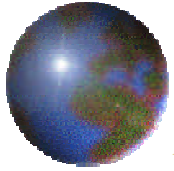
■ **Ability to scale up development**

- We now have middleware development taking place around the world rather than in one place.
- Being able to distribute integration development allows the integration team to be close to a large project no matter where it takes place.
 - SAP deployment in FL or JDE deployment in NJ.
 - We require consulting firms to use our methodology.

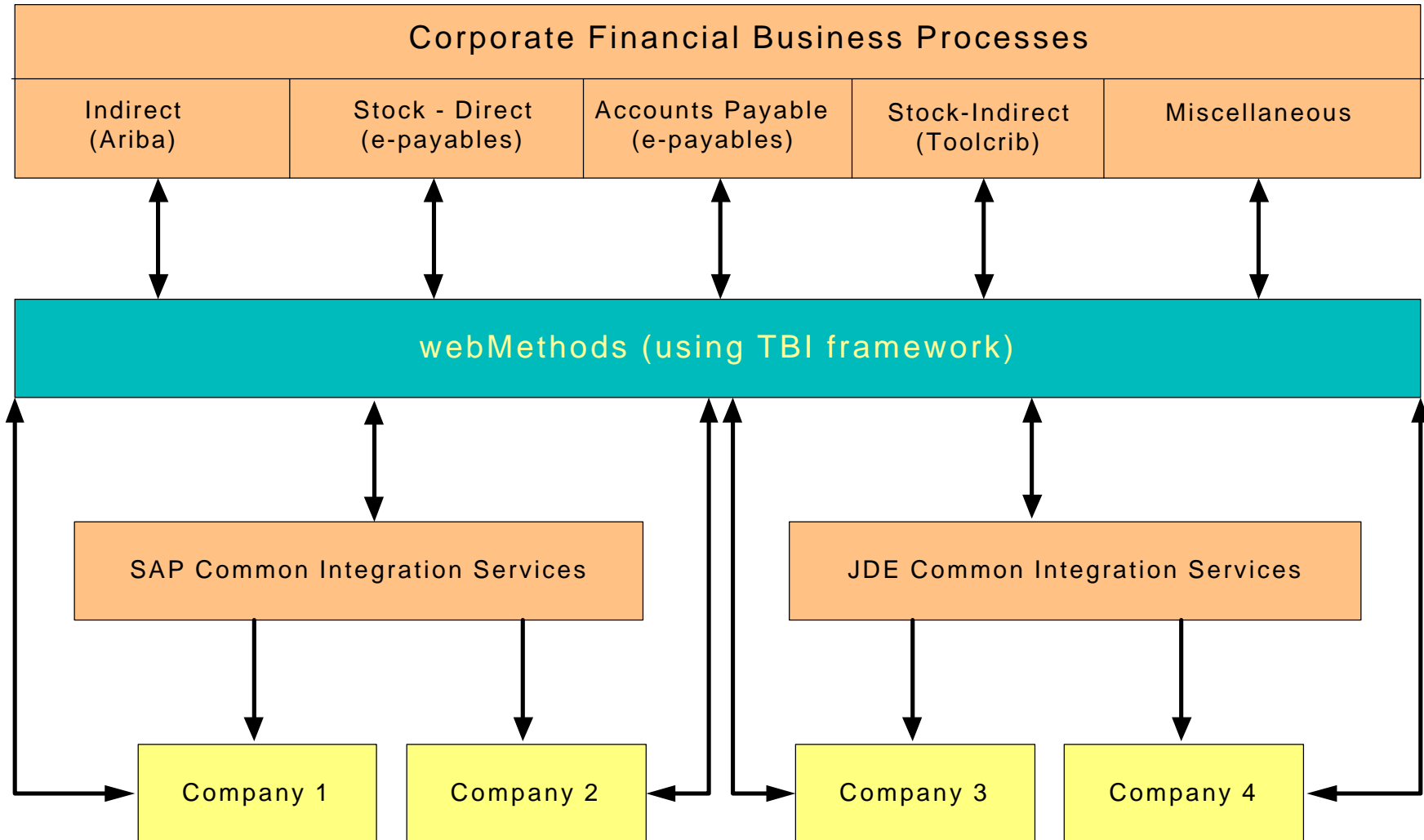


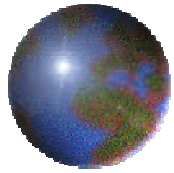
*Application of TBI to a large
integration project*

Ajay Anand

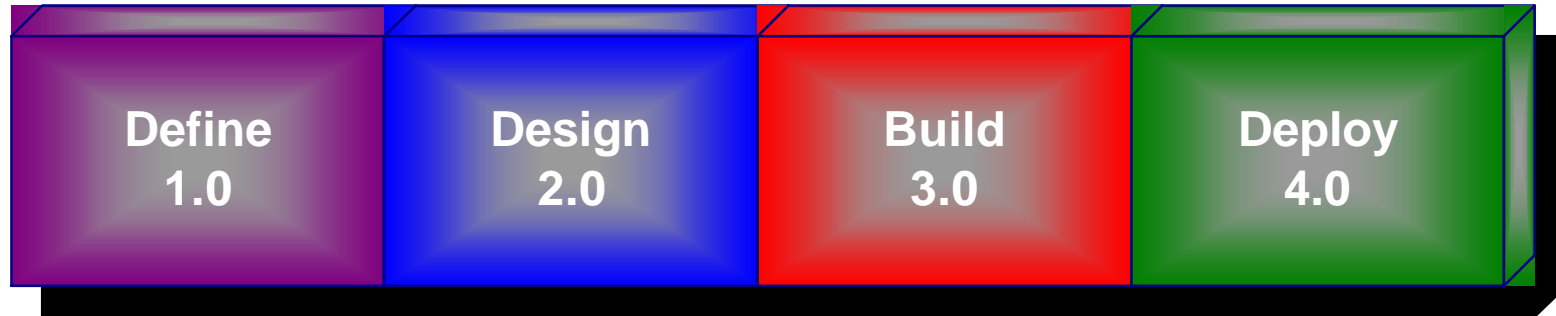


Procure-to-pay Integration





How TBI was applied ?

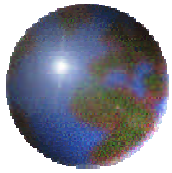


- Project Definition
- Business Process and Functional Areas
- SIPOC Diagrams
- Use Cases
- CTQ Document
- Technical Requirements
- SQA Plan and System Test Cases

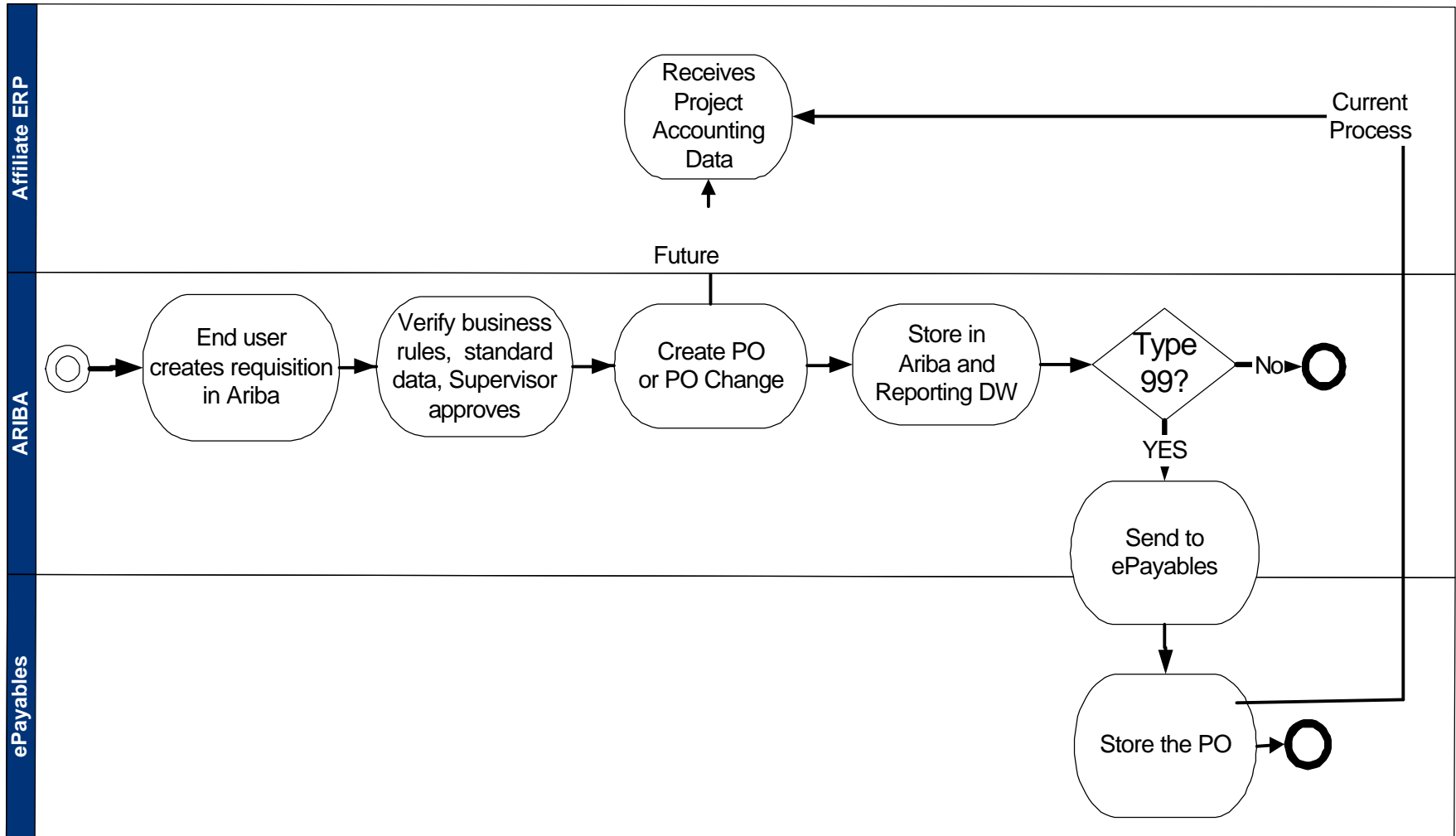
- Logical Design
- Integration Test Cases
- Architecture Document
- Simulation Document

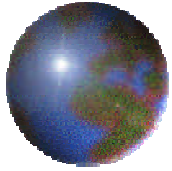
- Integration Design
- Source Code and Executables
- Documentation
- Unit Test Cases
- Test Results
- CTQ Acceptance

- Implemented Integration Solution
- Lessons Learned

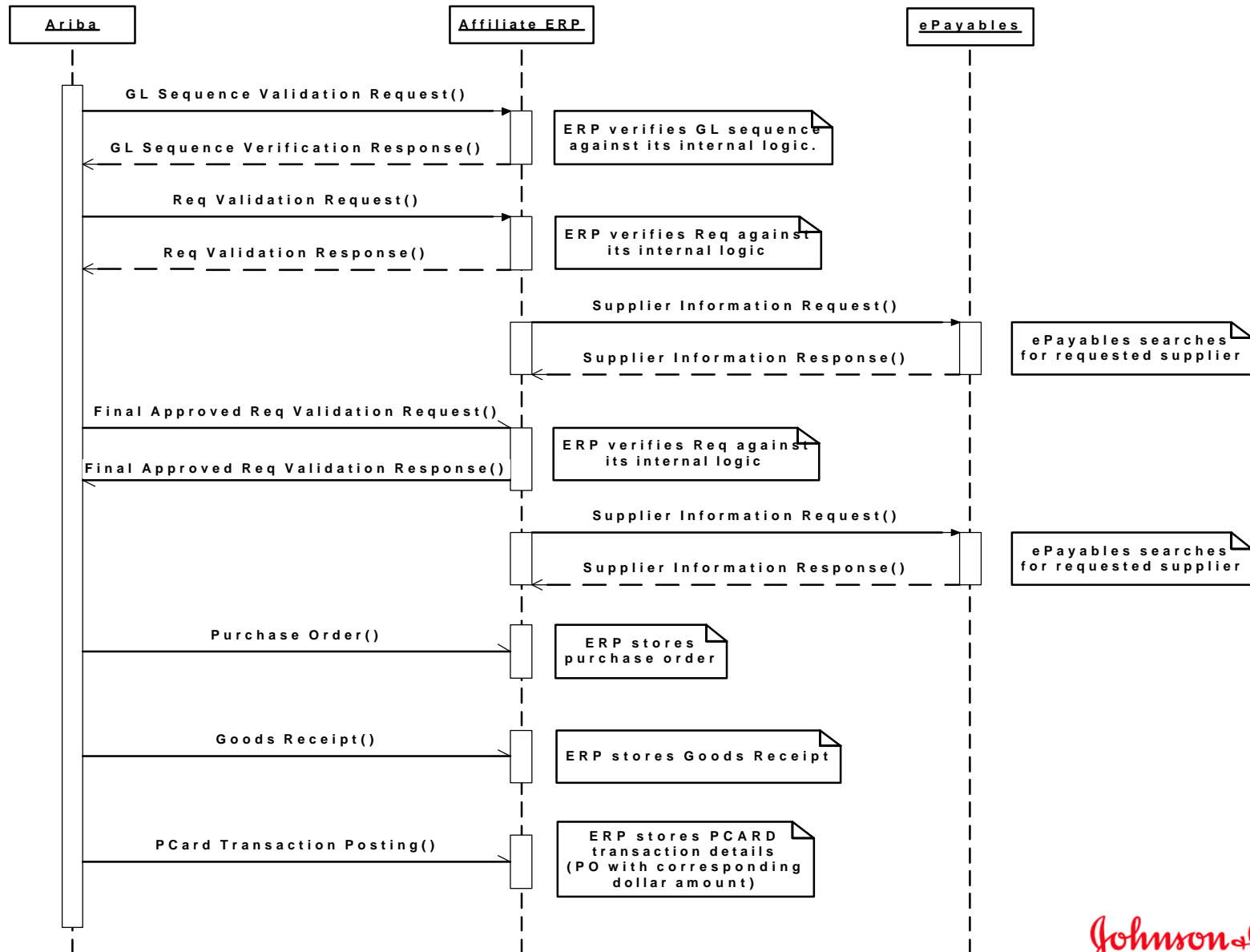


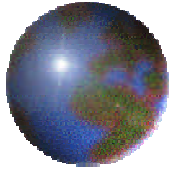
Business Process Analysis



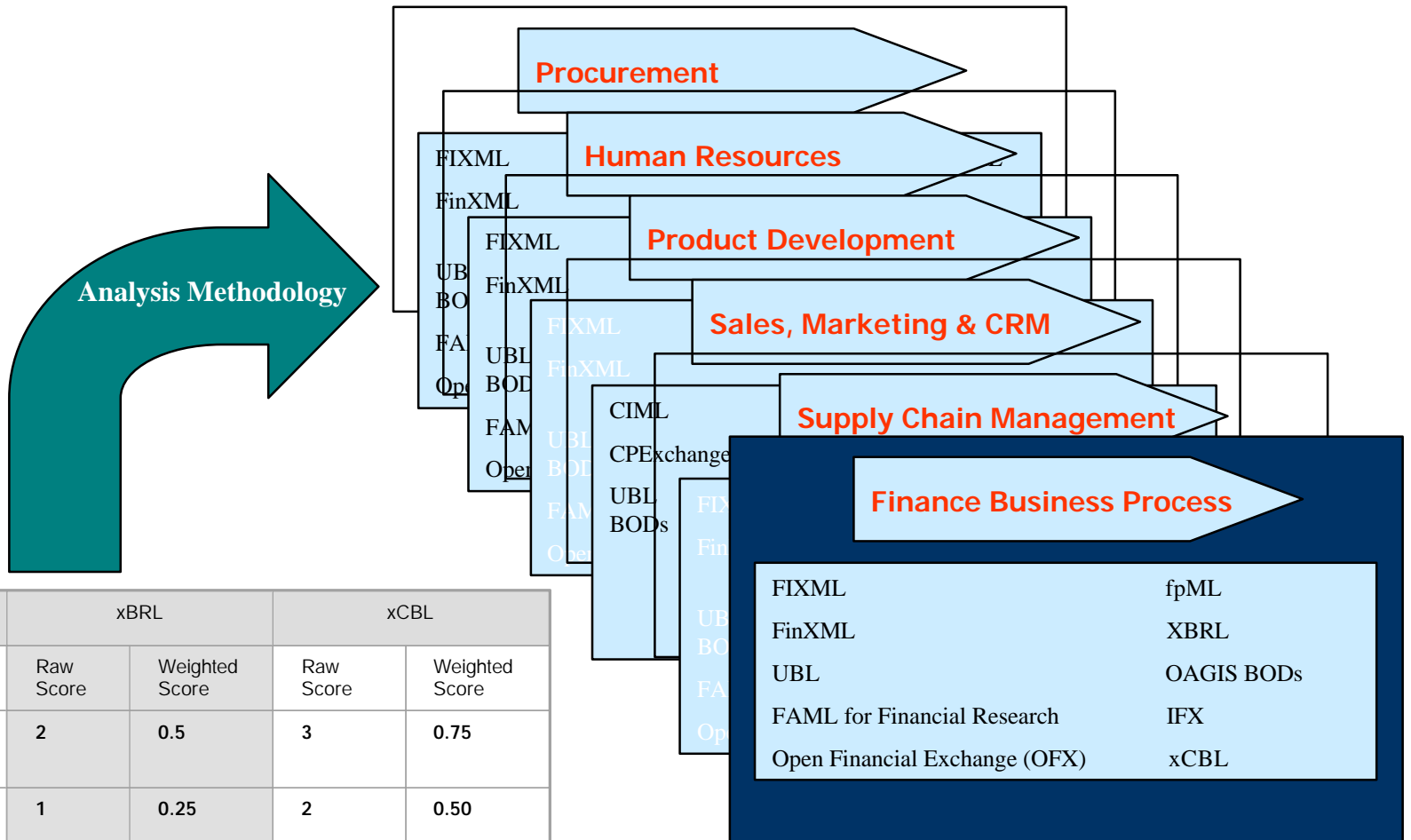


Sequence Diagram

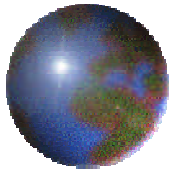




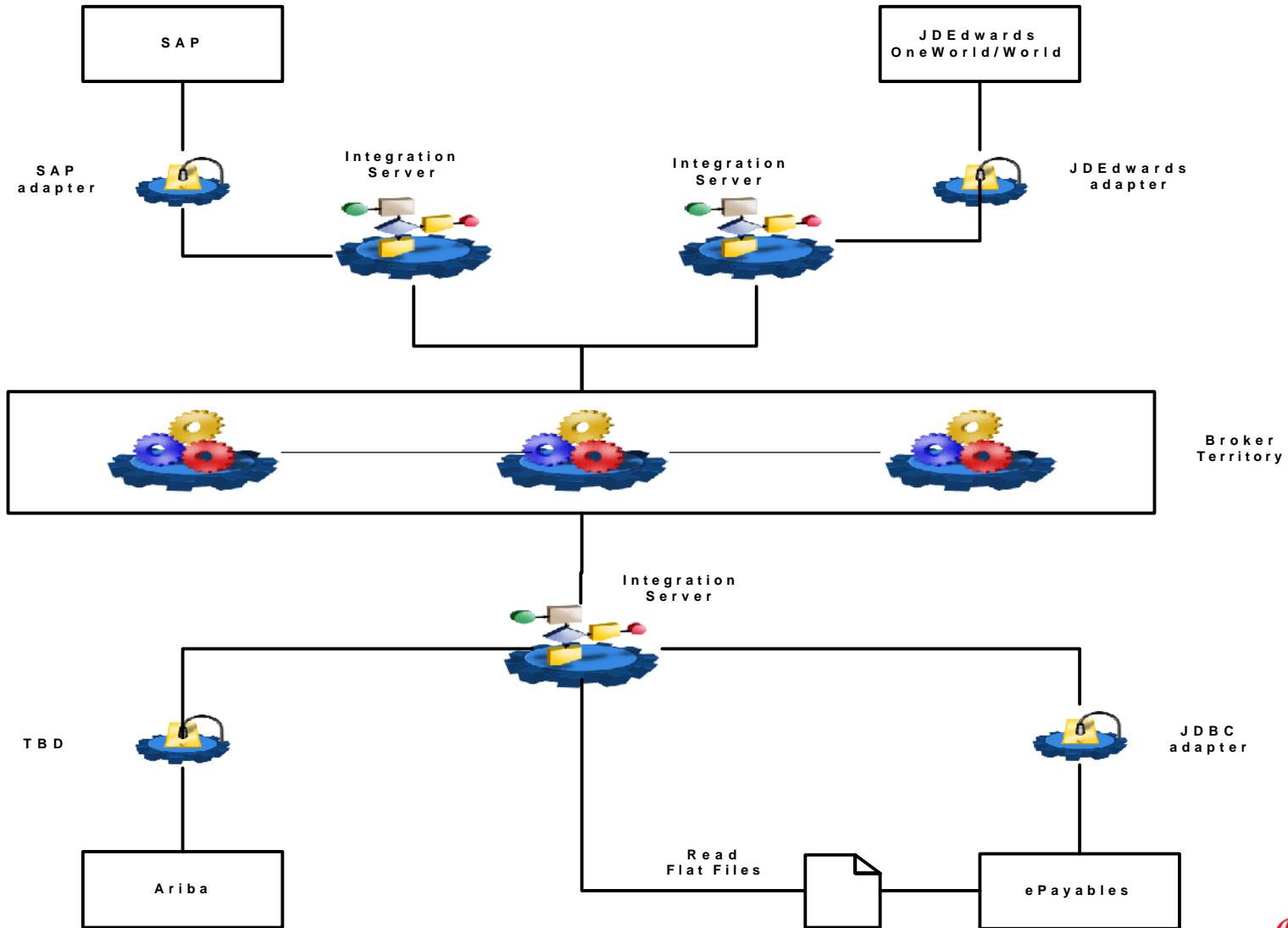
XML Standard Selection

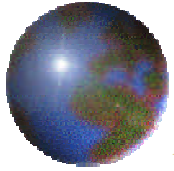


Criteria	xBRL		xCBL	
	Raw Score	Weighted Score	Raw Score	Weighted Score
Maturity & Industry Acceptance(25% Weight)	2	0.5	3	0.75
J&J Business Fit (25% Weight)	1	0.25	2	0.50
Technical Architecture (50% Weight)	2.1	1.05	2.3	1.15
Total	5.1	1.8	7.3	2.4



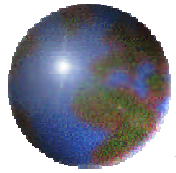
Conceptual Architecture





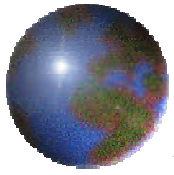
Architecture Recommendation

- **Architecture Analysis Document included:**
 - **Conceptual Architecture**
 - **Coordination Pattern For Component Communication**
 - **Application Communication Pattern Definition**
 - **Error Handling Approach**
 - **Architecture Review Approach**
 - **Security Considerations**
 - **Review of Infrastructure Needs**
- **Simulation was done to ensure that architecture meets customer's needs**

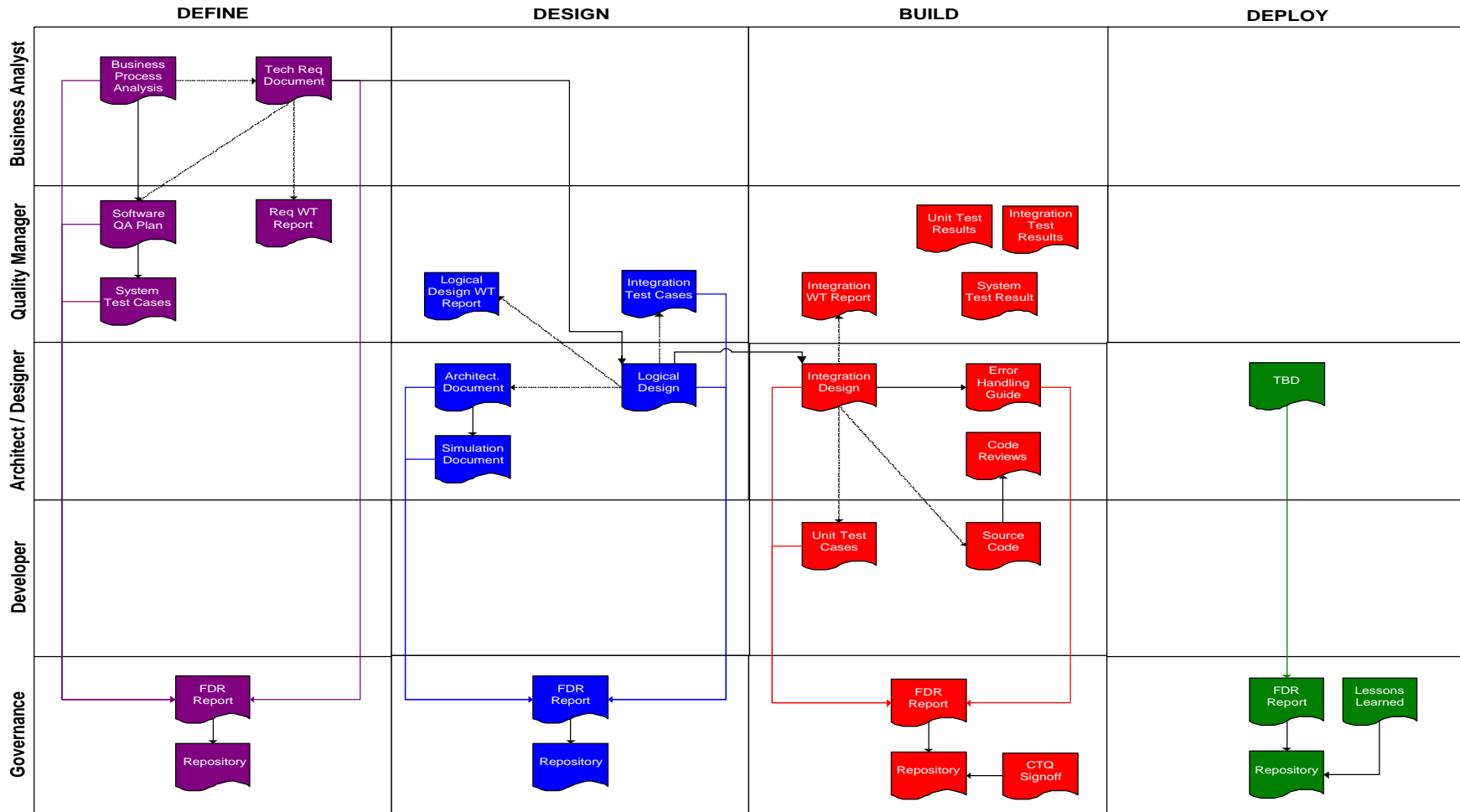


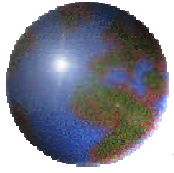
Build Activities

- **Integration Design** – Details the physical design of the interface point(s); includes naming standards, error handling, and security settings
- **Unit Test Cases** – Based on the integration physical design to ensure that the interface point adhere to the integration physical design
- **Source Code and Executables** – source code for the integrations and any executables (run-time code that may have been created
- **Code Review** – Summarizes the results, issues, and follow-ups that come out of a formal code review
- **Test Results** – Test Cases for unit, integration and system testing are all run in this phase; a summary is produced of all of the tests that were executed, and the results.



Deliverables Flow





Benefits from using TBI

- Reduced integration time & costs for initial and follow-on operating companies – 80% re-use goal (estimated savings for 6 companies above \$6 million)
- Standardized methodology across multiple companies (several sub-team's and SI's)
- Improved accuracy of project estimates
- Customer satisfaction
- Improved reliability
- Successful execution
- Simplified governance
- Lower TCO