



Lessons Learned in Software Requirements Management and Development 17 November 2006

DigitalFilipino

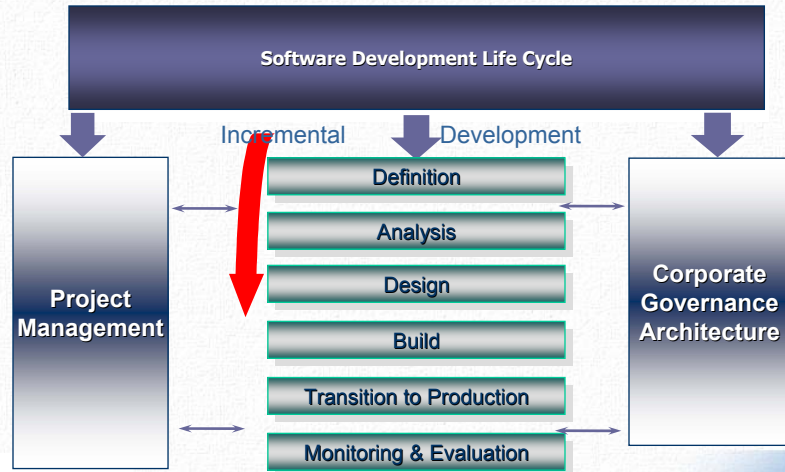


Agenda

- Corporate AVP
- Software Development Life Cycle
- Lessons Learned



Software Development Life Cycle




Lessons Learned

- Project Planning
 - Project Team and their Roles




Roles of Project Teams

- Executive Sponsor
 - Works with the project team to ensure the turnaround time on project issues or any related matters are minimized.
 - Ensures business users' compliance to project schedules
 - Approves and confirms major project decisions and changes that will affect project schedule, output and direction.
 - A representative shall be assigned in the project site whenever the Executive Sponsor is not available.
- 



Roles of Project Teams

- Project Manager
 - Should perform overall project management in terms of the following:
 - Business users interfacing
 - Gathering of requirements
 - Organizing meetings
 - Project status communications
- 




Roles of Project Teams

- **Steering Committee**
 - Will resolve major project issues concerning the organization
 - Standard members of the committee are the following:
 - Executive Sponsor
 - Project Manager / CIO
 - ISD Manager
 - Other members appointed by the organization
- 



Roles of Project Teams

- **Reference Group**
 - Involve in the following activities:
 - Intensive executive planning discussions
 - Project work regarding the organization's operations
 - System design proposals and process review
 - System testing activities for the validation of the system functionality and specification compliance
- 




Lessons Learned

- Project Planning
 - Project Team and their Roles
 - Requirements Definition
 - Keys to Excellent SW Requirements
 - Requirements Bill of Rights for Software Customers
 - Requirements Bill of Responsibilities for Software Customers
- 




Keys to Excellent Software Requirements

- Educating developers, managers, and customers about requirements engineering practices *and* the application domain
 - Establishing a collaborative customer-developer partnership for requirements development and management
 - Understanding the different kinds of requirements and classifying customer input into the appropriate categories
 - Taking an iterative and incremental approach to requirements development
 - Using standard templates for your vision and scope, use case, and SRS documents
 - Holding formal and informal reviews of requirements documents
 - Writing test cases against requirements
 - Prioritizing requirements in some analytical fashion
 - Instilling the team and customer discipline to handle requirements changes consistently and effectively
- 




Requirements Bill Of Rights for Software Customers

- ❑ Expect analyst to speak your language.
 - ❑ Expect analysts learn about your business and your objectives for the system.
 - ❑ Expect analysts to structure the requirements information you present into a software requirements specification.
 - ❑ Have analysts explain requirements work products.
 - ❑ Expect developers to treat you with respect and to maintain a collaborative and professional attitude.
 - ❑ Have analyst present ideas and alternatives both for your requirements and for implementation.
 - ❑ Describe characteristics that will make the product easy and enjoyable to use.
 - ❑ Be presented with opportunities to adjust your requirements to permit reuse of existing software components.
 - ❑ Be given good-faith estimates of the costs, impacts, and trade-offs when you request a requirement change.
 - ❑ Receive a system that meets your functional and quality needs, to the extent that those needs have been communicated to the developers and agreed upon.
- 



Requirements Bill Of Responsibilities for Software Customers

- ❑ Educate analysts about your business and define jargon.
 - ❑ Spend the time to provide requirements, clarify them, and iteratively flesh them out.
 - ❑ Be specific and precise about the system's requirements.
 - ❑ Make timely decisions about requirements when requested to do so.
 - ❑ Respect developers' assessments of cost and feasibility.
 - ❑ Set priorities for individual requirements, system features, or use cases.
 - ❑ Review requirements documents and prototypes.
 - ❑ Promptly communicate changes to the product's requirements.
 - ❑ Follow the development organization's defined requirements change process.
 - ❑ Respect the requirements engineering processes developers use
- 



Lessons Learned

- Documentation
 - Documentation Standards
 - Traceability
 - Prioritization
 - Baseline



THE END

