eFORCE Project Management Methodology

Aug 22, 2002

IT Solutions. Fixed Price. Fixed Time.™

Table of Contents

1. INTRODUCTION

As long as companies have been attempting to leverage Information Technology in their organizations, delivering business solutions to meet the goals of new or established businesses has been a major challenge. Today, IT solutions providers face an even more complex array of challenges in order to meet the requirements of building or enhancing Enterprise IT Solutions. Companies facing issues such as distributed computing technologies and legacy mainframe investments also need to factor in software quality, functionality, and budget constraints while addressing time-to-market goals. Competitive market forces and a constrained investment climate impose steep penalties for missteps. One compelling example is Nike, who announced in February 2001 that their flawed supply chain implementation prevented them from booking key orders during the quarter. The result: $2 Billion lost in market capitalization in a single day. Given this unforgiving climate, it is imperative that organizations execute flawlessly on IT initiatives and deliver them within the constraints of schedule, cost, functionality and performance.

While we have seen superior improvements in hardware and enabling software, the implementation track record of most software projects is not entirely comforting. The Standish Group[1], a leading IT research organization, published a study on the success rates of enterprise project implementations. The results are alarming. Less than 10% of these projects came in on time, on budget, or as planned. In addition:

- 35% of projects were cancelled.
- System functionality comprised only 57% of the original specifications.
- Projects had a cost overrun of 178%.
- Projects had a schedule overrun of 230%.

Further, projects that could be considered reasonably successful had the following common characteristics:

- Heroic efforts were required to finish the project.
- Schedule and cost were compromised.
- Quality standards were not achieved.
- The solution involved a fair amount of rework.

1.1. Why does one need a methodology?

In an industry where management and technical personnel often view tools and technologies as a silver bullet, this is a very legitimate question. eFORCE believes that in no case should a methodology be revered as ”the be-all and the end-all” or panacea, but instead it should serve the following important purposes:

- Methodology serves as a common lexicon amongst all the key stakeholders—end-users, technology providers, business owners, executives, etc.

[1] http://www.standishgroup.com/ChaosU/

- Methodology allows junior personnel to become more effective and senior personnel to become more efficient.
- Well-designed methodology serves as a common roadmap where progress can be monitored and measured against pre-determined metrics.

1.2. What should a methodology involve?

- A methodology should be reasonably comprehensive, i.e. it should cover the key aspects of the conceived solution; however, it should not attempt to be all encompassing.
- It needs to be thorough and rigorous but should not be overly-constraining.
- It should provide flexibility for the range of different projects a company might undertake.
- It should enforce discipline but not take away critical judgment and insight.
- It should leverage commonly-used industry standard notations (example: UML – Unified Modeling Language) and tools (RUP - Rational Unified Process) rather than trying to invent new approaches.
- It should provide a clear communications vehicle to all parties involved in the project – business stakeholders, analysts, programmers, architects, and managers.
- , Most importantly a methodology should allow all the key stakeholders (technology provider, business owner, financier, etc.) to develop increasing clarity on the key aspects of the project (scope, schedule, price, performance) and mitigate risk through the successive project phases.

2. Project Critical Success Factors - The 4 Ps

eFORCE believes that it is imperative to address the following "4 Ps" of successful project management.

People: eFORCE Project Managers, Business Architects, Domain Specialists, Technical Architects and Engineers are chosen with very rigorous standards. In addition to rigorous screening, they go through formal training/certification and mentoring by more experienced personnel.

Product: eFORCE bases its solutions on proven, field-tested products with robust technical architectures, requisite functional capabilities and compliance with industry standards. Enabling technologies from vendors who meet strict eFORCE screening criteria ensures that the products and the solutions based on these represent low implementation risk and acceptable cost of ownership.

Place: eFORCE operates Centers of Excellence globally in Hayward, CA; Los Angeles, CA; Dallas, TX; New York, NY; Boston, MA; London, UK; Calcutta, India; and Chennai, India. These centers have prototyping labs, facilitation rooms, appropriate development hardware/software, and networking/security equipment, and are staffed by certified professionals. This allows eFORCE to deliver strategic IT solutions without the latency that is typically involved in most projects.

Process: Most approaches in the industry straddle the extremes of brute force development and theoretical and impractical techniques espoused by the academic community. eFORCE's methodology is based on practical, flexible, industry-standard approaches that leverage the 'best practices' experience of practitioners from the industry.

In addition to applying the non-proprietary RUP methodology, eFORCE leverages its expertise in deploying over 200 solutions at Global 1000 organizations and mid-tier companies to evolve and institutionalize best practices in order to deliver "Fixed Time, Fixed Price" and Quality solutions.

In addition, eFORCE project personnel leverage:

- Practical industry experience derived from delivering over 200 solutions at leading Global 1000 organizations
- Best practices approaches defined by industry organizations like the Project Management Institute (www.pmi.org)
- Approaches espoused by leading industry pioneers such as Steve McConnell[2] , Fred Brooks[3], Ed Yourdon[4], and Watts Humphrey[5].

eFORCE has a stated goal of achieving 100% customer satisfaction. One key aspect to achieving customer satisfaction is for the client and the technology provider to subscribe to a shared vision of the project—and a process to achieving

[2] Steve McConnell, IEEE Software, March 1997

[3] Project Manager of the IBM 370 Operating System and author of the legendary book "Mythical Man-Month"

[4] Pioneer of Structured Design

[5] Watts S. Humphrey pioneer of the CMM (Capability Maturity Model), Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA

the project goals. RUP serves as this common navigational tool. The following sections outline in greater detail the different phases in the eFORCE project process.

3. eFORCE Methodology

eFORCE's development methodology ***is*** the Rational Unified Process. Our methodology is not "based on" or "modified from" the standard—it is the standard. This provides the framework that allows our teams to deliver complex systems in a manner that is timely, cost-effective and predictable.

3.1. Why RUP

Most Systems Integrators tend to extol the benefits of their proprietary methodologies. While eFORCE could have chosen this route and in fact did so several years ago, we now believe that there is tremendous value for clients in a non-proprietary methodology. Our reasoning is as follows:

- RUP is a well-documented, public, and de-facto industry standard. Unlike propreitary methodologies, RUP will continue to grow in value as the global community continues to contribute to the base of best practices.
- RUP uses UML (Unified Modeling Language) notation, an open industry standard (http://www.omg.org/uml/) that is endorsed by over 30 leading software vendors
- RUP complements existing enterprise project management methodologies
- RUP has broad industry support in terms of tools, techniques, best practices templates
- RUP-standard methodology averts project delays and cost overruns due to semantic gaps by imposing a standard lexicon of project-related terms
- RUP facilitates incremental, iterative and parallel development, an approach with proven cost benefits
- RUP gives customers ultimate flexibility—Vendor independence

3.1.1. eFORCE Value Add

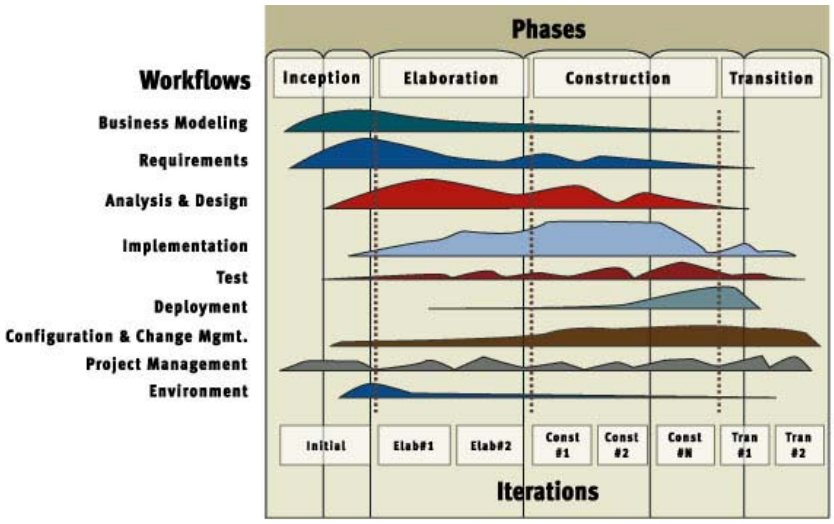
Leveraging the de-facto standard, RUP, the following are unique eFORCE value add:

- RUP is a very comprehensive methodology that involves a broad range of approaches, processes and tools. Because of its expertise from over 200 solutions, eFORCE can assist clients in selecting the optimum, most efficient and most relevant sub-set from the entire RUP methodology
- eFORCE has developed "best practice" RUP templates/artifacts for its specialized practices. Examples include:
  - Enterprise Content Management (ECM) Systems
  - Enterprise Portals
  - Customer Relationship Management (CRM) Systems
  - Enterprise Application Integration (EAI)
  - J2EE migration from legacy architectures
  - High Performance Architectures – Delivering QoS (Quality of Service) in business critical applications
  - Global Delivery Methodology (http://www.eforceglobal.com/pdf/whitepapers/eF.WP-eGDM_07-12-02.pdf)
- eFORCE ensures that projects have clearly-defined start and end points, and that as many issues and risks as possible are identified up front.

Thus, unlike the approach of other vendors who deliver solutions based on proprietary methodologies, eFORCE's approach is to leverage de-facto standard approaches and add value to them.

We understand that while process and methodology are necessary components to successful execution, they are not sufficient to ensure success. Methodology provides predictability and repeatability; it is collective experience and best practices and superior talent that actually execute. The Rational Unified Process provides explicit guidance in areas such as business modeling, change management, schedule management, architectures, and testing. It is eFORCE's superior expertise that converts these things into real functionality.

The diagram below depicts the four phases of the rational unified process.



In systems solution deployments, Cost, Time to Market, Functionality and Quality are strategic vectors. In an effort to meet the project schedule, a number of systems initiatives have overlooked critical aspects of solution deployment. This has resulted in a number of publicized failures—applications that are not scalable, are too costly to maintain, or simply cannot meet the functional and technical requirements.

eFORCE leverages process and people to deliver "Business Critical" solutions in rapid timeframes, in a manner that is straightforward, predictable and comprehensible by our customers. Even though we are committed to open standards in process and technology, we are able to gain a competitive advantage in the marketplace using practical techniques and pragmatic approaches that our competitors do not. Some of them include:

- Prioritized and phased implementations of functionality grouped in deliverable milestones that are as frequent as is practical.
- Facilitated group requirements sessions that align stakeholders and staff with the project methodology.
- Highly interactive and incremental prototypes used to capture and test user-system interactions in a tangible manner.
- Development activities that are both parallel and iterative.
- Disciplined UML design artifacts for business logic, transaction architecture, object models, database designs, and deployment architectures.
- Meticulous testing and quality assurance from elaboration through transition—testing is elaborated concurrently with technical design and continues to be applied throughout the life of the system.

- Technology transfer to customers. We believe that the best quality implementation is one where our customers want to retain us as their vendor of choice, but do not need to do so.
- Formal project tracking and reporting.
- Single point-of-project ownership within the customer and eFORCE organizations.

The sections that follow trace eFORCE-developed applications through a typical RUP lifecycle. The phased approach ensures that both eFORCE and our clients gain increasing clarity over the project's key aspects: scope, schedule, and cost.

3.2. eFORCE Use of RUP

The phases associated with RUP are illustrated below in Figure 1. RUP divides projects into four overlapping phases. Activities within each phase are defined with specific deliverables (artifacts) as that serve as input to the next phase.

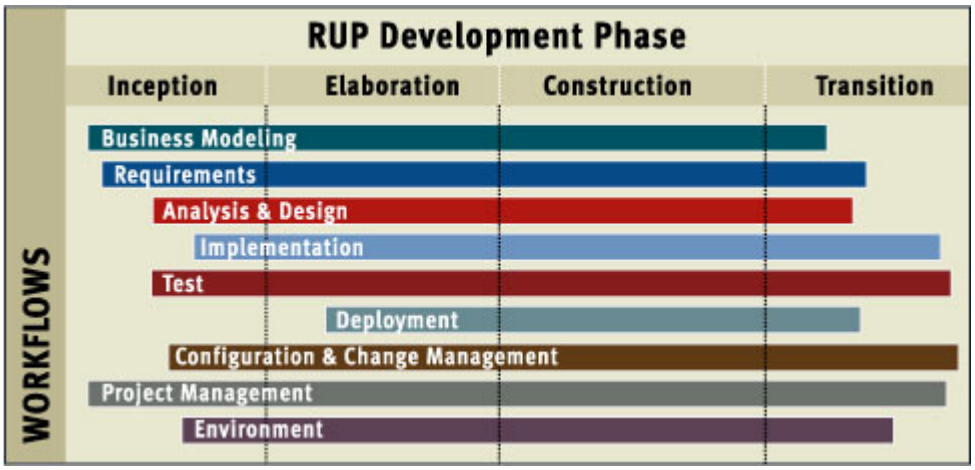


Figure 1

The RUP phases are summarized below:

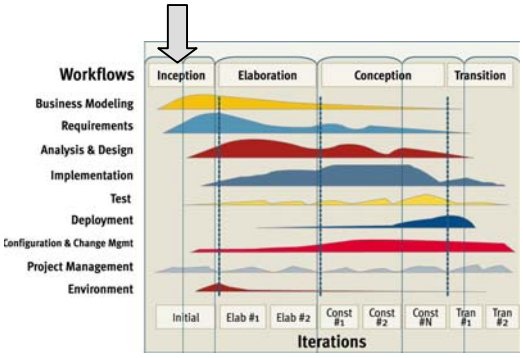
- **Inception Phase**—Baselines the solution framework,functional components, technical architecture, priorities, schedule, test strategies, acceptance plans and defines a task plan, milestones, staffing plan and costs for the following phases.
- **Elaboration Phase**—Develops and documents the architecture of the system to provide a stable basis for the design and implementation effort in the construction phase.
- **Construction Phase**—Clarifies the remaining requirements and completes development of the system using the documented architecture.
- **Transition Phase—**Ensures that the software is meets the functional, stability, performance and availability requirements of its end users. This phase can span several iterations, and includes testing the product in preparation for release, making minor adjustments based on user feedback.

The following sections detail each of the 4 RUP Phases.

3.2.1. Inception Phase

Milestone Objectives

- Analyze business needs and document baseline requirements
- Develop business case
- Design prototype

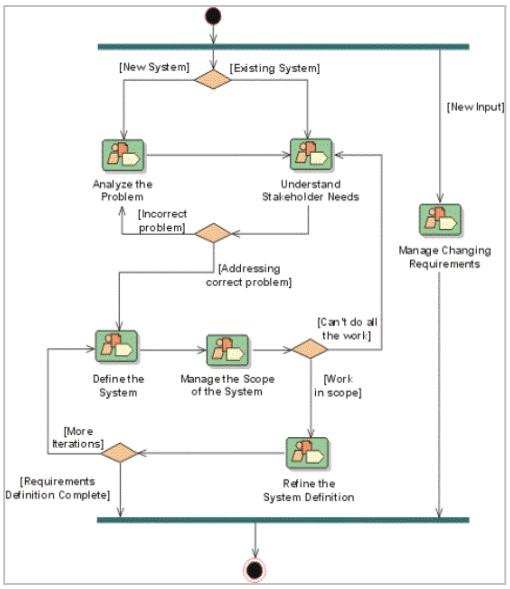


Milestone Evaluation Criteria

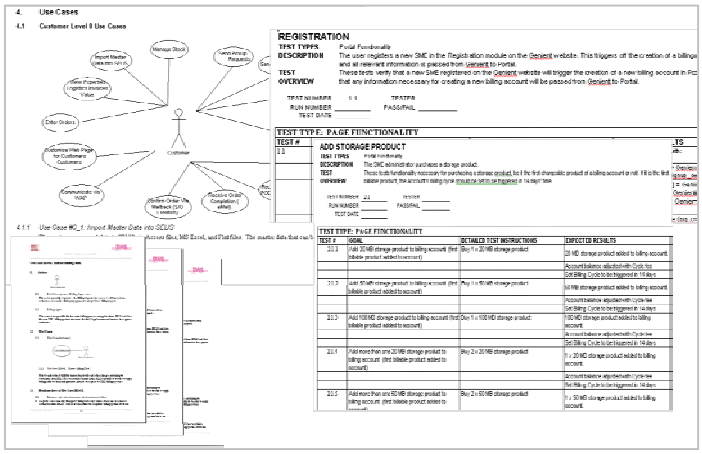
- Stakeholder concurrence on scope definition and cost/schedule estimates
- Agreement that the right set of requirements has been captured and that there is a shared understanding of these requirements
- Agreement that the cost/schedule estimates, priorities, risks, and development process are appropriate
- All risks have been identified and a mitigation strategy exists for each

Essential Activities

- Develop/Document Business Case
  - Identify business drivers
  - Identify technical drivers
  - Cost of ownership and ROI assessment
  - Benefits assessment t
- Develop storyboards to define the system
- Analyze Business Needs and document Requirements
  - Refine system definition based on Gap Analysis and document extension requirements
  - Prioritize and catalogue all functional and non-functional requirements, such as performance, availability and security
  - Identify key actors and use cases
  - Analysis of Quality of Service Requirements and initial Capacity Planning (user load, response time, user adoption rate, etc.)
  - Document product acceptance, software development and iteration plans
- Risks assessment
- Identification of initial audience
- High-level project scope and timelines
- Project funding/sponsorship
- Design Prototype



Key Artifacts



- Business Case & Vision: The Business Case is defined and approved. The prioritized system requirements, key features and main constraints are documented.
- ROI Assessment Worksheets: To supplement the business case, if needed.
- Risk List: Initial project risks identified.
- Software Development Plan: Initial phases, their duration,objectives and major milestones identified. Resource and cost estimates for the entire project (±20%) or just the elaboration phase (±5%).
- Iteration Plan: First Elaboration Iteration only.
- Product Acceptance Plan: Reviewed and baselined; refined in subsequent phases as more requirements discovered.
- Use-Case Model: Important actors and use cases identified and flow of events outlined for the most critical use cases.
- High Level Page Flow Model: Outline of the key screen to screen navigational flow.
- Glossary: Important terms defined; glossary reviewed.
- Prototype: One or more proof of concept prototypes to support the vision and business case and to address very specific risks.
- Development environment, Test Environment and Production environment configuration and support plan.
- Project management system support (file structure, naming conventions) is complete.

Key Exit Critera

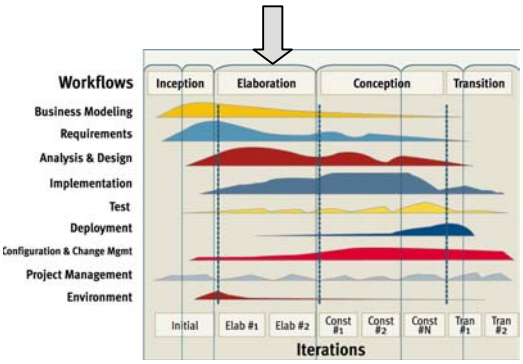
- eFORCE has a firm understanding of who the business client is an their specific business processes.
- eFORCE has a understanding of Client's vision and how it relates to what they want in the project.
- eFORCE has a understanding of what the business requirements are in order for this project to be sucessful.
- EFORCE and Client have a firm agreement on Client resource commitments and project dependencies.
- eFORCE has a understanding of the technical requirements required for this project to be successful.
- eFORCE and the Client have mutually reviewed their understanding of the business and technical needs.

- eFORCE and Client have determined project success criteria.
- eFORCE understands the client's current infrastructure and what the apparent hardware needs are in the recommened solution.
- Client has made the necessary decisions concerning the purchase of middleware, hardware, etc.

3.2.2. Elaboration

Milestone Objectives

- Complete system design
- Create a base configuration
- Set up change/configuration management

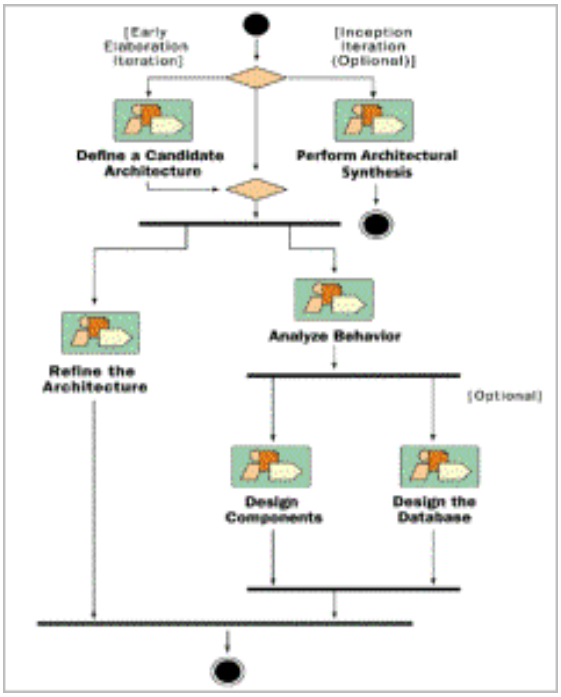


Milestone Evaluation Criteria

- System vision, requirements and architecture are stable.
- Prototypes have demonstrated that major risks have been addressed.
- The iteration plan for construction phase are of sufficient detail to allow work to proceed and are supported by credible estimates.
- Stakeholder concurrence that current vision can be met if current plan is executed as per current architecture.
- Actual resources expenditure versus planned expenditure is acceptable.

Essential Activities

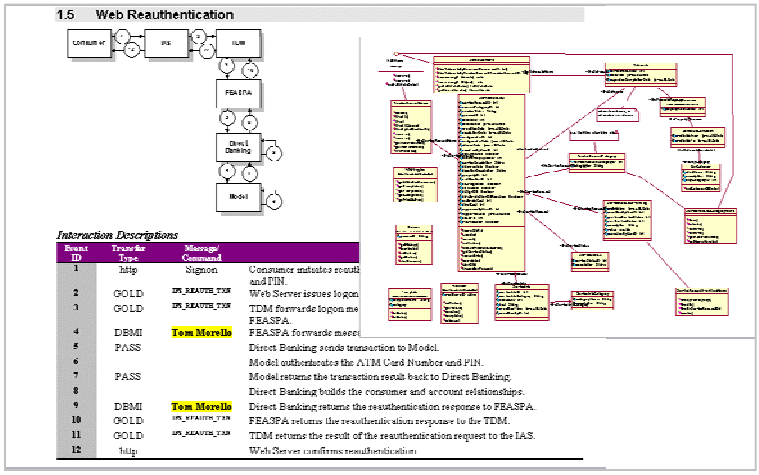
- Complete System Design
  - Develop technical architecture document.
  - Detailed technical design including software, hardware, network and security architectures.
  - Develop object and data models corresponding to the extension requirements.
  - Develop interface specifications for all integrations.
  - Update software development plan, iteration plan and use-case model.
  - Develop Quality of Service (QoS) capacity model. Identify QoS Capacity Risks and design QoS Tests.
  - Identify all major risks and develop mitigation strategies.
  - Develop additional prototypes if necessary.



- Create a Development Environment Base Configuration
  - Set up infrastructure & tools to support development
  - Set up the database repository & load sample data
  - Set up developer security
  - Set up Code Management System
  - Verify the base configuration
  - Start up server and address any errors
  - Set Up Change/Configuration Management
  - Set up source code control tool
  - Domain and Server files
  - Enterprise application and Web application files
  - Change control process
  - Change requests, approval workflow

Key Artifacts

- Functional Requirements: Finalized and approved. Completed Functional Requirments have been reviwed and approved by key stakholders.
- Prototypes: One or more executable architectural prototypes to explore critical functionality and architecturally significant scenarios.
- Risk List: Updated and reviewed. New risks are likely to be architectural in nature primarily relating to the handling of non-functional requirements.
- Technical Architecture Document: Detailed technical architecture including network, hardware, software, security design.
- Design Model: Defined and baselined. Includes use-case realizations for architecturally significant scenarios. Object/component model and integration interface definition/specifications.
- Logical Data Model: Defined and baselined. Major data model elements (entities, relationships, tables) defined and reviewed.
- Software Development Plan: Updated and expanded to cover the Construction and Transition phases.
- Iteration Plan: Updated and expanded to cover the Construction phase to include initial code build schedule.
- Use-Case Model: Approximately 80% complete. All actors and use cases identified and their



descriptions captured.

- QoS Capacity Model and QoS Test Plan.
- Development Environment configuration is completed.
- Test Straegy: Define the overall testing strategty for the system, idetifying resources, process, test tools, data and change control.

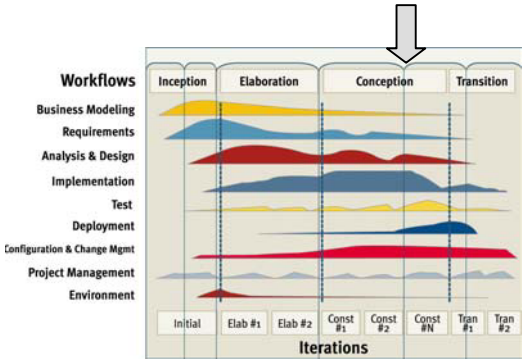
Key Exit Critera

- The architecture specification (creative, business and technical) is detailed enough to begin the Construction phase.
- The detailed software requirements specificiation (creative, business, technical) has been completed.
- The detailed design documentation (creative, business, technical) has been completed.
- The engineering specification of the busines problem has been completed.
- The low-level project plan for the Development phase has been completed.
- Development and QA enfironments have been defined.

3.2.3. Construction

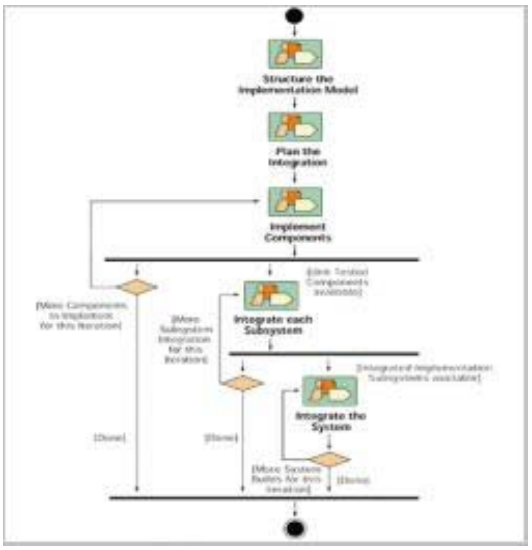
Milestone Objectives

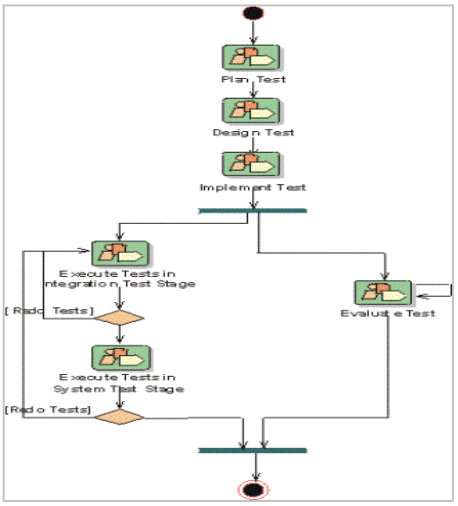
- Set up and configuration of development site(s)
- Set up and initailize basic services and environments
- Set up personalization, commerce and campaign services
- Perform system development
- Integrate with the extended enterprise
- Develop and build componets based on the iteration and build plans.Deploy iterations in a test environment
- Conduct System & Integration Testing



Key Activities

- Configure System Test Environment
- Allow for concurrent development
- Organize tasks into discrete units, e.g. promotional campaign, webflow modification, etc.
- Use source control to manage the data between separate instances and the master server
- Similar set up for all application source code and content
- Keep an audit trail/history of changes - Authentication/authorizations
- Integrate with the Extended Enterprise - CRM, ERP, SCM, Mainframe systems integration
- Deploy in a Test Environment
- Set up integration with external test systems
- Build and deplpoy physcial database in test development environment
- Conduct System and Integration Testing
  - Execute all system and integration test plans
  - Implement "continuous" testing during development based on iterations
  - Execute all scripts every day, even multiple times if possible
  - Each iterative release is unit, system and integration tested
  - Automate system and integration testing if possible using standard tools - WinRunner, LoadRunner, JTest, etc.
- Validate against user acceptance criteria and other system specifications.
- Create QoS (performance, scalability, reliability) test scripts
- Measure Quality of Service (QoS) against QoS requirements for each major system build.
- Tune system performance and iterate measurements.
- Implementation of QoS monitoring instrumentation for QoS Dashboard (optional).



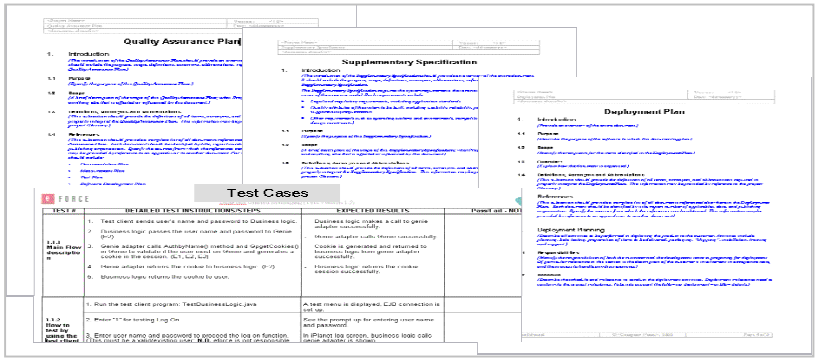


Milestone Evaluation Criteria

- Are all the stakeholders ready for the transition into the user community?
- Are all unit and functional test complete and approved
- Are actual resources expenditures versus planned expenditures still acceptable?

Key Artifacts

- "The System": The deployed platform, fully integrated and ready to start system testing, is released in successive iterations.
- Test Plan: Includes functional, integration and user acceptance testing.
- Deployment Plan: Initial version developed, reviewed and baselined.
- Test Model: All tests designed and developed to validate system through construction iterations.
- Iteration Plan: Updated and expanded to cover the Transition phase.
- Design Model: Updated with new design elements identified during the completion of all requirements.
- Physical Data Model: Updated with all elements needed to support the persistence requirements (tables, indexes, object-relational mappings).
- Database: Physical database configured, built and production data migrated.
- Supplementary Specifications: Contains all non-functional requirements, e.g. performance, QoS, availability, scalability, etc.
- Use-Case Model: Updated with any use cases discovered during the construction phase.
- QoS Capacity Measurement report.



Key Exit Critera

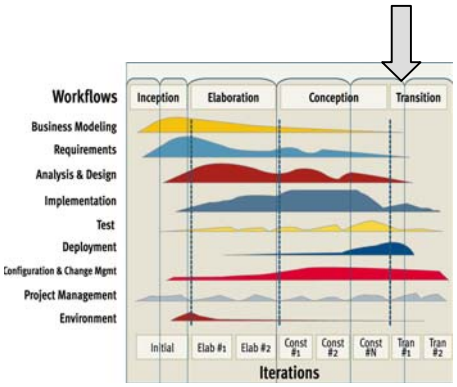
- The Implemenation Plan has been created and approved.
- All code has been completed and unit tested.
- Any necessary graphics have been created and approved.
- All accepted change orders have been documented and forwarded to the client's Test Lead.
- All Elaboration documentation has been updated to reflect the changes made to the design as a result of the change control process.
- The Traceability Matrix is updated to reflect design changes.
- The Construction phase quality review had been performed.

- The Deployment plan has been created and submitted for approval.
- The Data Migration plan has been created and submitted for approval.
- The Code documentation has been completed.
- All Unit Testing results have been forwarded to the client for review.
- Performance metrics have been created and compared against those guidelines described in the Elaboration documentation, results are forwarded to the client, and any actual metrics that differ from the guidelines have been corrected.
- The Software release/version control document has been created and approved.
- The plan for User Acceptance Testing (UAT) has been created and approved.
- All docuementation in the following areas has been created:
  - Technical
  - Development Environment
  - Security
  - Administration
  - End User

3.2.4. Transition

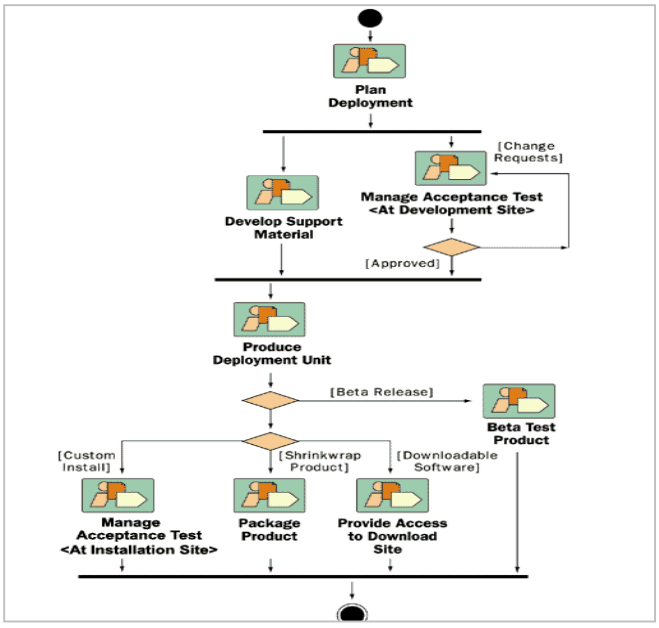
Milestone Objectives

- Execute full-system functional, performance, scalability, and exception testing.
- Perform final user acceptance testing
- Ensure smooth cut over to production
- Address key post-deployment issues such as systems operations, technology transfer, training, maintenance, support, etc.
- Work with stakeholders to evaluate ROI measures and ensure deployed system meets the stated objectives of the business case.



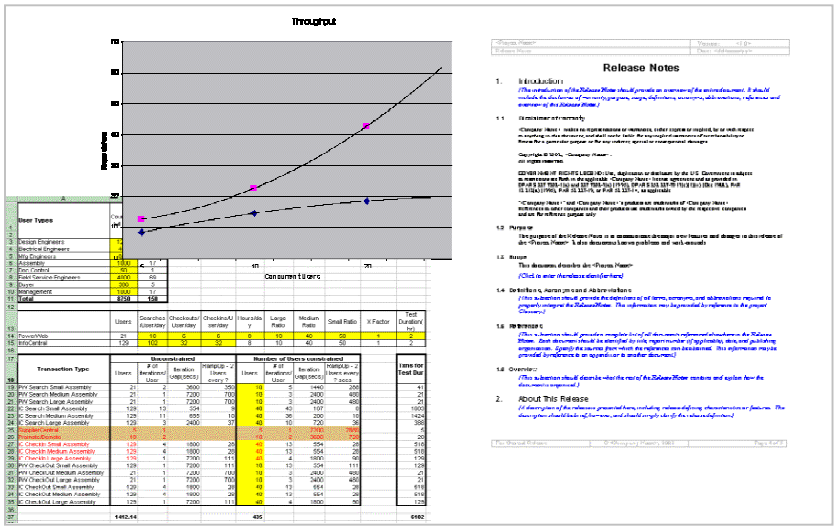
Milestone Evaluation Criteria

- Is this system release stable and mature enough to be deployed in the user community?
- Are all stakeholders and user representatives satisfied?
- Does the system meet the stated objectives of the business case?
- Are actual resources expenditures versus planned expenditures still acceptable?
- Has all operational system knowledge been transferred to the maintenance personnel?
- **Production Environment Deployment**
- Tune system performance and iterate measurements
- Measure and monitor production system resource capacity limits
- Document system performance tuning and maintenance procedure
- Set up system Quality of Service (performance, scalability, reliability) dashboard
- Deploy in the Production Environment
- Set up production infrastructure
- Firewalls, load balancers, web servers, application server clusters, RDBMS servers, file servers, etc.
- Install 3rd party software
- Set up integrations with external production systems
- Set up/verify system security such as SSL certificates, authentication & authorization



Key Artifacts

- "The System Build": Complete in accordance with all system requirements. The final release should be usable by the customer
- Release Notes: Complete
- Installation and maintenance documentation: Complete
- Training Material: Complete to ensure that the customer can become self-sufficient in the use of the system
- Client/Operations Support Material: Complete to ensure that the customer can become self-sufficient in the use of the system
- System QoS Capacity Model: For on-going scaling of the system as user load increases.



Key Exit Critera

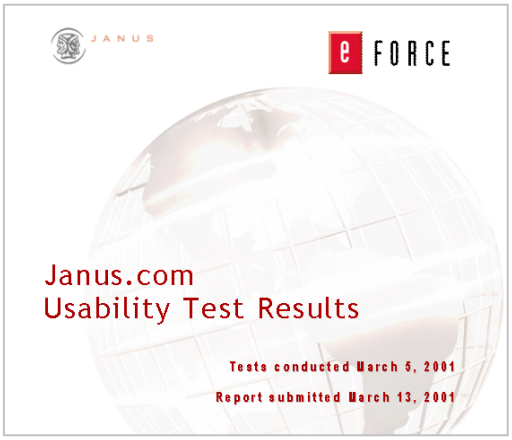
- Release of vendor patches have been assessed
- All compatibility issues have been identified (hardware, software, interface) and resolved
- Firewall issues have been identified and resolved
- Feature requests for the next release have been identified in the defect/bug tracking and reproting system and have been removed from this release.
- The usability results and strategy have been identified and the effect on this release have been determined.
- Staging procedures have been defined for the production environment, tested with the client, and any issues have been resolved.
- All links that have been tested on all the pages, including static pages, are accessible
- The migration of the legacy data has been completed, tested and all issues have been resolved.
- All integration points have been completed and tested and all issues have been resolved
- All DNS issues, based on any changeses that occurred during production rollout, have been defined and resolved. The final DNS switchover has been established.
- All SSL issues have been resolved
- All load testing and stress testing issues have been resolve and are meeting required metrics for go-live. A plan is in place for the next adjustments and/or upgrades.
- All training needs have been defined, client's team are being trained on schedule and all traning issues have been resolved.
- A determination has been made on whether or not there is a need for HPA services
- Post-launch support and all of its tasks have been defined.

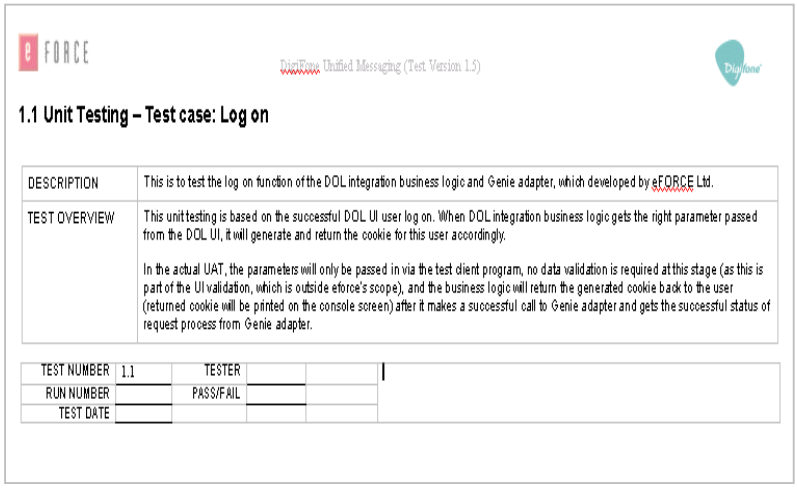
3.3. eFORCE Quality Approach

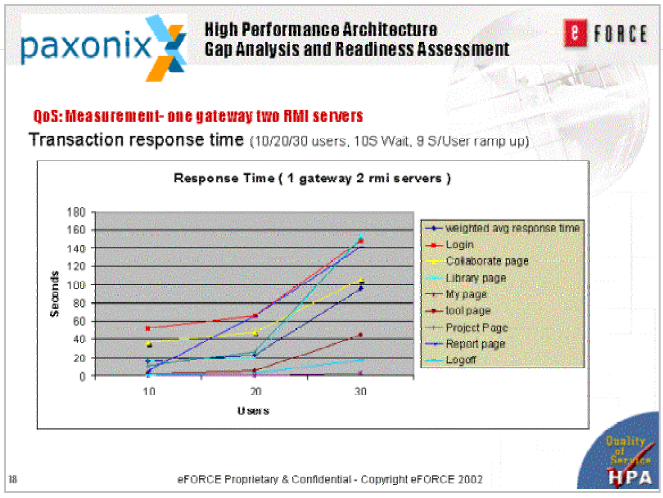
Fundamental to eFORCE's quality disciplines, is that functionality, Quality of Service (performance, scalability and reliability) and usability should be a part of the engagement from the project's inception. eFORCE implements quality disciplines throughout the project lifecycle. The following table outlines the set of functional and performance quality checkpoints addressed as a part of the project throughout the entire project timeline.

Inception	Elaboration	Construction	Transition	Post-deployment Maintenance
▪ Define and Analyze Quality Assurance Requirements ▪ Define and Analyze User Acceptance Criteria ▪ Define and Analyze Quality of Service Requirements ▪ Define and Analyze Human Factors Requirements.	▪ Design Quality Assurance Plan ▪ Design Quality Assurance Environment ▪ Define Communication and Data Transfer Protocols ▪ Develop Coding Standards and Guidelines ▪ Design Defect Management and Reporting	▪ Implement Quality Assurance Infrastructure ▪ Implement Defect Management and Reporting ▪ Implement Test Suites ▪ Execute Quality Assurance on Builds ▪ Measure and Analyze Quality and Defect Management	▪ Design, Analyze Implement Deployment Plan ▪ System Integration Testing and Reporting ▪ Data Migration testing ▪ QA of Production Deployment Environment ▪ QA of Security Policy ▪ QA of System Documentation and Training Material. ▪ Manage Documentation Transfer and "Hand Off"	▪ Defect Resolution Maintenance ▪ System Quality of Service Monitoring, Measurement and Analysis ▪ System Load and Scalability Reviews ▪ On-going Defect Management and Reporting Post Deployment

3.3.1. Sample Artifacts (Functional, Performance, Usability)







3.4. Post Deployment

eFORCE offers its clients a variety of post-deployment support options. They include:

- Technology transfer and mentoring
- Upgrades and updates
- Functional enhancements
- Extended warranty periods
- Remote monitoring
- On-site or off-site maintenance and support, from 24X7 coverage to support on an as-needed basis

Unlike packaged software vendors who offer a standard set of support services, the post-deployment options that eFORCE offers its clients are tailored to suit the specific needs of the client.

4. eFORCE Methodology- Summary Template

INCEPTION	ELABORATION	CONSTRUCTION	TRANSITION
TASKS/ACTIVITIES ▪ Vision/goals/objectives for the project and development of a business case ▪ Shared view between stakeholders and vendor on: ▪ High level scope ▪ Schedule ▪ Budgets ▪ Performance ▪ Review of 'As Is' system Documentation ▪ Design Prototype	**TASKS/ACTIVITIES** ▪ Detailed discussion with: ▪ Technical users ▪ Business users ▪ Infrastructure personnel ▪ Create a base configuration ▪ Implement Proof of Concepts to mitigate key risks ▪ Complete system design with design reviews ▪ Set up change/configuration management	**TASKS/ACTIVITIES** ▪ Set up a development site ▪ Set up portal services ▪ Integrate with the extended enterprise ▪ Development with daily software builds ▪ Weekly Integration Builds ▪ Frequent Code Reviews ▪ Quality Assurance ▪ Unit Testing ▪ System Testing ▪ Integration Testing	**TASKS/ACTIVITIES** ▪ Full-system functionality, performance, scalability and exception testing ▪ User Acceptance Testing ▪ Production Deployment ▪ Technology Transfer ▪ Training ▪ Gradual Phase-Out of Legacy Code Base
ARTIFACTS ▪ Vision document ▪ Business case ▪ ROI Assessment worksheets ▪ Risk list ▪ Software development plan ▪ Iteration plan ▪ Product acceptance plan ▪ Use-case model ▪ Glossary ▪ Prototype	**ARTIFACTS** ▪ Prototypes ▪ Risk list ▪ Technical architecture document ▪ Design model • Object/Class model • Sequence diagrams • Interaction diagrams ▪ Data model ▪ Software development plan ▪ Iteration plan ▪ Use-case model	**ARTIFACTS** ▪ Iterative system builds ▪ Code base ▪ Database scripts ▪ Test model and results ▪ Deployment plan ▪ Iteration plan ▪ Design model ▪ Data model ▪ Supplementary specifications ▪	**ARTIFACTS** ▪ The fully integrated Portal system ▪ Full test results ▪ System capacity model and performance test results ▪ Release notes ▪ Training material ▪ Client support material

5. Project Monitoring, Controls and Risk Management

eFORCE institutes a variety of project monitoring and change control mechanisms to ensure that projects remain on track. At the highest level we maintain a change management process around the original statement(s) of work and master agreements. This ensures that changes made at this level are properly cascaded through the impacted project(s). During requirements definition we use traceability tools and techniques to track each requirement through the development and testing processes. Within our development efforts we maintain a rigorously managed code library for each project, which allows us to incorporate changes to the design quickly and with predictable results.

eFORCE recognizes that each client can have unique needs and goals. In order to meet these needs eFORCE works with clients at the very beginning of every engagement to gain a common and shared definition of the success criteria. Based upon the defined criteria we construct a project plan and a set of deliverables that will clearly demonstrate progress towards these criteria. eFORCE also recognizes that these criteria may change through the life of a project, and we have mechanisms in place to allow for such changes to be incorporated in the delivery plan. In general, we recognize that the dominant criterion for success in a technology deployment is an on-time, on-budget delivery that meets the client's business needs and quality of service goals. By these measures eFORCE leads our competitors by a significant margin.

5.1. Risk Management

The process of developing and deploying "business critical" IT solutions with stringent ROI (Return on Investment) goals is a significant and risky undertaking. Many technology implementations and application development projects exceed their original budget and time schedule goals. Projects generally fail due to inadequate project management, attention, commitment and prioritization.

eFORCE believes that the prudent approach to managing and executing complex systems integration and reengineering projects is to recognize and closely control the risks inherent in these types of projects, so that there are no surprises. To address this need, eFORCE has developed a risk mitigation process as part of our standard project support that outlines the purpose, process and risk mitigation guidelines as a means of providing effective risk management assistance focused on our client's priorities of cost and schedule control. This is how we have been able to achieve and maintain our 85% on-time, on-budget delivery track record.

The purpose of risk mitigation is to manage and control the risk of project deficiencies that might cause unanticipated cost and schedule overruns. The following are the primary risks that this process is designed to help manage:

- Scope creep
- Mismanagement of user expectations
- Unfulfilled expectations of project executive sponsors
- Application software functionality shortfalls
- Hardware technology inadequacies, e.g., capacity, capability, etc.

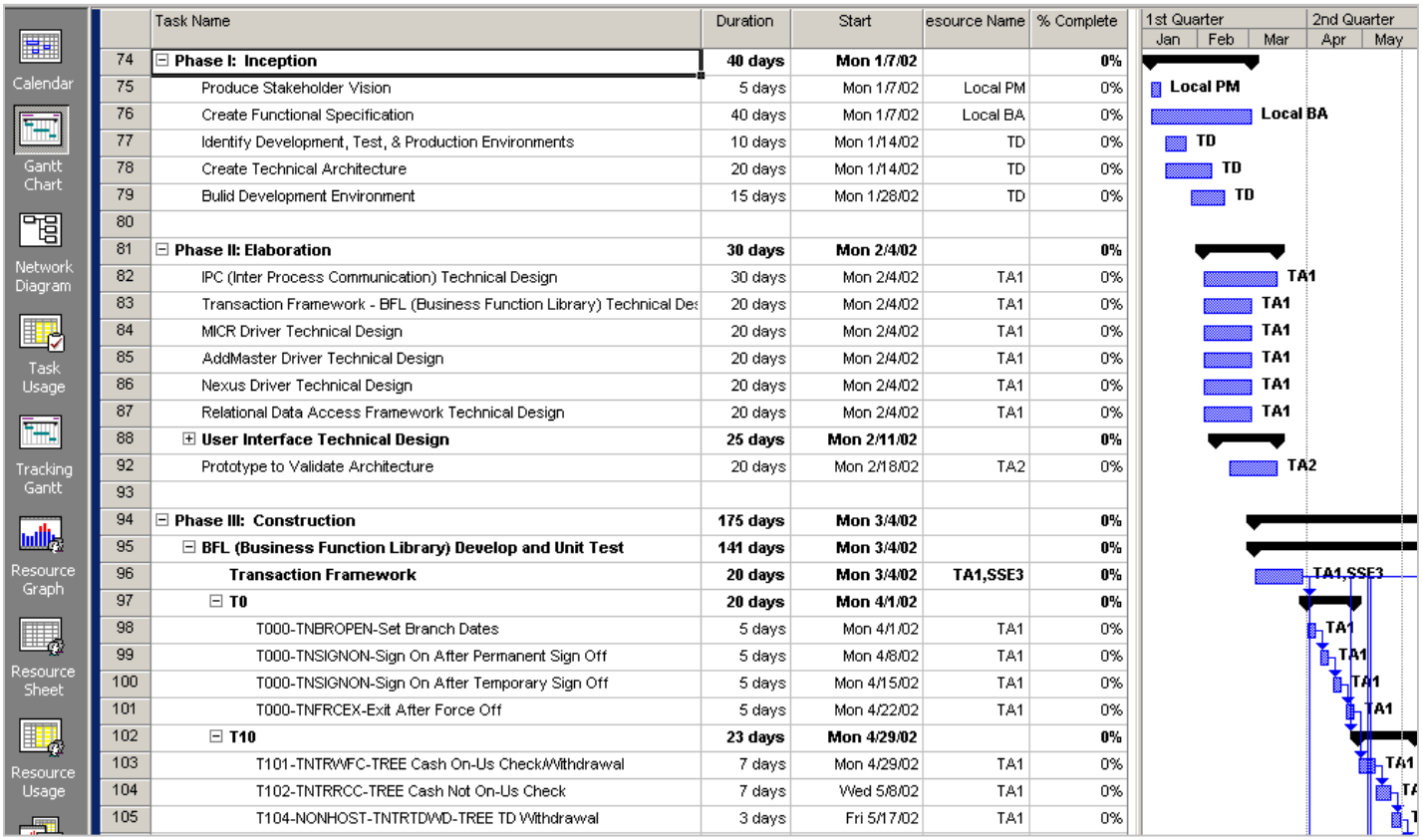
- Incorrectly scoped, unidentified, and/or missing application software interfaces
- Vendor under-performance, missed commitments, or loss
- Lack of clear accountability, responsibility and control for performance
- Inadequate staff resources
- Lack of management commitment
- Lack of priority for task completion
- Lack of user understanding of benefits and project ROI
- Lack of performance measurement process for measuring benefit realization
- Loss of key personnel
- Interruption of Development/Test environment

Management of project risk will be the responsibility of all team members. The eFORCE Project Manager will have the direct responsibility to determine whether the risk mitigation process has appropriate compliance. The Project Manager will report on a monthly basis to the eFORCE Advisory Team who will review the project for compliance. In addition to the Advisory Team review process, risk management will be accomplished in the following manner:

- **Release Planning**: Evaluation of the most effective way to package and build releases to minimize deployment risk. Effective release planning allows for the placement of well-defined checkpoints to evaluate organizational readiness and technology effectiveness incrementally.
- **Quality of Performance**: Review of the expected service and performance level agreements during the initial phases to set proper expectations and to ensure that performance and scalability are managed and designed in all components of the system architecture.
- **Executive Expectations and Commitment**: Interviews with executive sponsors early in the project aid understand and help manage expectations with respect to project objectives, scope and resource requirements.
- **Scope Expansion**: Continual focus on controlling project scope expansion, including the areas of functionality, organizations, business processes, technology, vendors and project staffing.
- **Project Management Process**: Formal weekly work plan analysis by the Project Management Team and the manager.
- **Requirements Traceability**: All requirements are to be catalogued and numbered for tracking purposes. Catalogued requirements define project scope and must be signed off by the appropriate parties.
- **Technical Risk Management**: Determine whether technical risk is being managed with subject matter specialist's assistance.
- **Change Management:** Determine whether an effective change management program is in place for implementation activity.
- **Risk Management Plan:** Plan for non scope and un-controlable risks, by documenting mitigation plans for common risks including the loss of key personel, loss of devopment/test environments for extend period, etc.

5.2. Project Plans

Project Plans are the schedule for the whos, whats, and whens of the project. The eFORCE project manager maintains the plan that tracks the project's tasks, dependencies, and resources that are performing these tasks. Here is a representative view of a typical project plan.



	Task Name	Duration	Start	esource Name	% Complete
74	**Phase I: Inception**	**40 days**	**Mon 1/7/02**		**0%**
75	Produce Stakeholder Vision	5 days	Mon 1/7/02	Local PM	0%
76	Create Functional Specification	40 days	Mon 1/7/02	Local BA	0%
77	Identify Development, Test, & Production Environments	10 days	Mon 1/14/02	TD	0%
78	Create Technical Architecture	20 days	Mon 1/14/02	TD	0%
79	Bulid Development Environment	15 days	Mon 1/28/02	TD	0%
80					
81	**Phase II: Elaboration**	**30 days**	**Mon 2/4/02**		**0%**
82	IPC (Inter Process Communication) Technical Design	30 days	Mon 2/4/02	TA1	0%
83	Transaction Framework - BFL (Business Function Library) Technical Des	20 days	Mon 2/4/02	TA1	0%
84	MICR Driver Technical Design	20 days	Mon 2/4/02	TA1	0%
85	AddMaster Driver Technical Design	20 days	Mon 2/4/02	TA1	0%
86	Nexus Driver Technical Design	20 days	Mon 2/4/02	TA1	0%
87	Relational Data Access Framework Technical Design	20 days	Mon 2/4/02	TA1	0%
88	**User Interface Technical Design**	**25 days**	**Mon 2/11/02**		**0%**
92	Prototype to Validate Architecture	20 days	Mon 2/18/02	TA2	0%
93					
94	**Phase III: Construction**	**175 days**	**Mon 3/4/02**		**0%**
95	**BFL (Business Function Library) Develop and Unit Test**	**141 days**	**Mon 3/4/02**		**0%**
96	**Transaction Framework**	**20 days**	**Mon 3/4/02**	**TA1,SSE3**	**0%**
97	**T0**	**20 days**	**Mon 4/1/02**		**0%**
98	T000-TNBROPEN-Set Branch Dates	5 days	Mon 4/1/02	TA1	0%
99	T000-TNSIGNON-Sign On After Permanent Sign Off	5 days	Mon 4/8/02	TA1	0%
100	T000-TNSIGNON-Sign On After Temporary Sign Off	5 days	Mon 4/15/02	TA1	0%
101	T000-TNFRCEX-Exit After Force Off	5 days	Mon 4/22/02	TA1	0%
102	**T10**	**23 days**	**Mon 4/29/02**		**0%**
103	T101-TNTRWFC-TREE Cash On-Us Check/Withdrawal	7 days	Mon 4/29/02	TA1	0%
104	T102-TNTRRCC-TREE Cash Not On-Us Check	7 days	Wed 5/8/02	TA1	0%
105	T104-NONHOST-TNTRTDWD-TREE TD Withdrawal	3 days	Fri 5/17/02	TA1	0%

5.3. Status Reports

It is critical that project status is reviewed on a weekly basis to ensure that the project is on track. eFORCE produces a status report for the weekly status meeting and uses the following agenda to make sure that the project stays on schedule:

- Review previous week's key accomplishments
- Review key future project milestones and deliverables
- Review project issues in the issues list
- Review project risks in the risk plan

Below is a sample project status report:

Project: **FFE (Fiduciary Functionality Expansion)**
Week: **5/13 to 5/20**
Time: **10:00 AM Mountain Time**

1. Milestones - % Complete (Please see FFE Project Plan – 20000520.mpp)

Milestones	% Complete
Inception Phase	100%
Elaboration Phase	98% (Publishing Final Documentation)
Construction Phase	3% (Started)
Transition Phase	0%

3. Key Accomplishments last week (5/13/02- 5/20/02)

- Commenced presentation (jsp) and middle tier (dispatcher, controller, business object) development.
- Conducted the mid point project review at eFORCE. Discussed the project to date, technical designs, HTML Mockups, and construction/transition plan. Updated the issues and risk lists.
- AMO is merging into PVCS. Working through issues including:
  - Delivering builds with files locked. The development team does not have PVCS manager privileges and cannot open those files.
  - Security for PVCS was not pushed with the archive
  - Waiting for the merge results

4. Key Upcoming Dates (FFE Project Plan – 20000520.mpp)

- 5/16 – Final copy approved and delivered
- 6/3 – Test data requirements are provided to Corporate Testing
- 6/7 – Test plans delivery and review 1 (review first set)
- 6/14 – Test plan delivery and review 2(review second set)
- 6/14 – DST Beta API Installation

5. Project Issues:

- Issues can be found in the project issues list (FFE Issue List 20020520.xls).

6. Project Risks:

- Issues can be found in the project risk plan. (FFE Risk Plan 1.8.doc)

5.4. Issue Tracking

Issues are defined as those items that could alter the quality, cost or schedule of the development project. Issues are important to address and resolve as soon as possible, as neglecting them can detrimentally affect delivery. Issues are tracked and updated at least every week, and form a key part of the weekly status report.

Here is a representative view of the eFORCE standard Issue Log:

	A	B	C	D	E	F
1	**General**					
2	**Issues**	**Owner**	**Date Opened**	**Due Date**	**Status**	**Resolution**
24	eFORCE has install the iAS and iWS successfully but is waiting for some data files to set up a identical environment as Janus production env. Some of the files are: LDIF file for LDAP server	Frank / Mike Ryan	5/14/2002	5/24/2002	Open	Steve Pfleeger is the first escalation point after 5/17
25	Is Ptarmagin going to be able to support both IPSS and FFE testing at the same time?	Steve	5/17/2002	5/24/2002	Closed	The FFE project has already expressed solid dates. In Ram's mind we have priority. We do have the facility.
26	Is Dave Korn available for testing or brain storming?	Steve	5/17/2002	5/24/2002	Open	We can have that session without Dave. Maybe Melanie can be involved. We want to make sure that we do not miss any back end output testing cases.
27	We need to establish the number of testers from the number of test	Mike	5/21/2002	6/7/2002	Open	

5.5. Risk Tracking

Critical risks are tracked separately from the project issues. The risks that are tracked represent the significant items that would impact the delivery of the project. These risks are reviewed on a weekly basis in the project status meeting so that all parties are aware of critical dependencies and project risk.

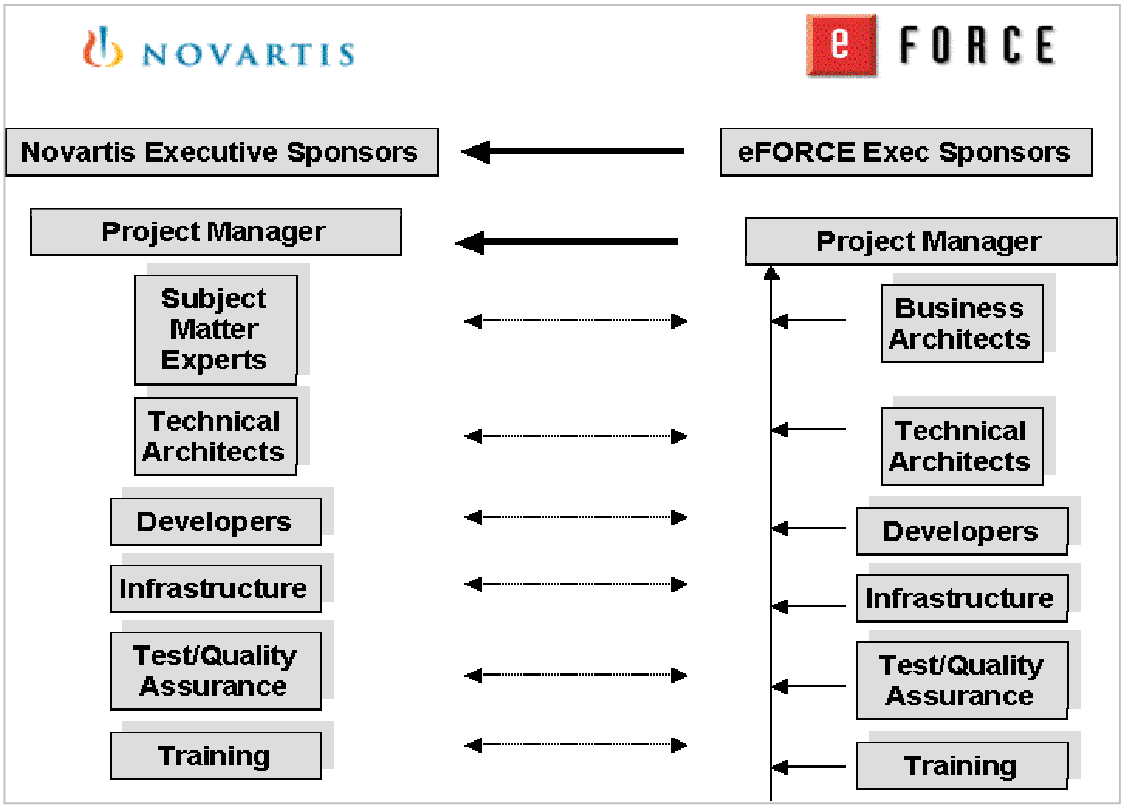
5.6. Action Item Tracking

Action Items are tasks that arise during a project that are related to resolving an issue. Most project tasks are defined through the project plan, where timeframes, resources and dependencies are assigned. However, tasks related to the resolution of short- or long-term issues are typically below the task time threshold for project planning (usually less than 4 hours duration) and are frequently changing in nature, making them difficult for a project plan to track effectively.

These action items are typically tracked directly in the Issues log. This makes it clear who owns the issue, what action is to be taken, and what prior actions have been completed.

5.7. Project Organization

eFORCE believes that project organizations have to be defined in an unambiguous fashion at the outset of the project. In addition, eFORCE believes that there is significant benefit in combining the benefits of a hierarchical organization while leveraging the benefits of a peer to peer relationship that is critical for effective and low latency communication. Such a project organization is shown below.

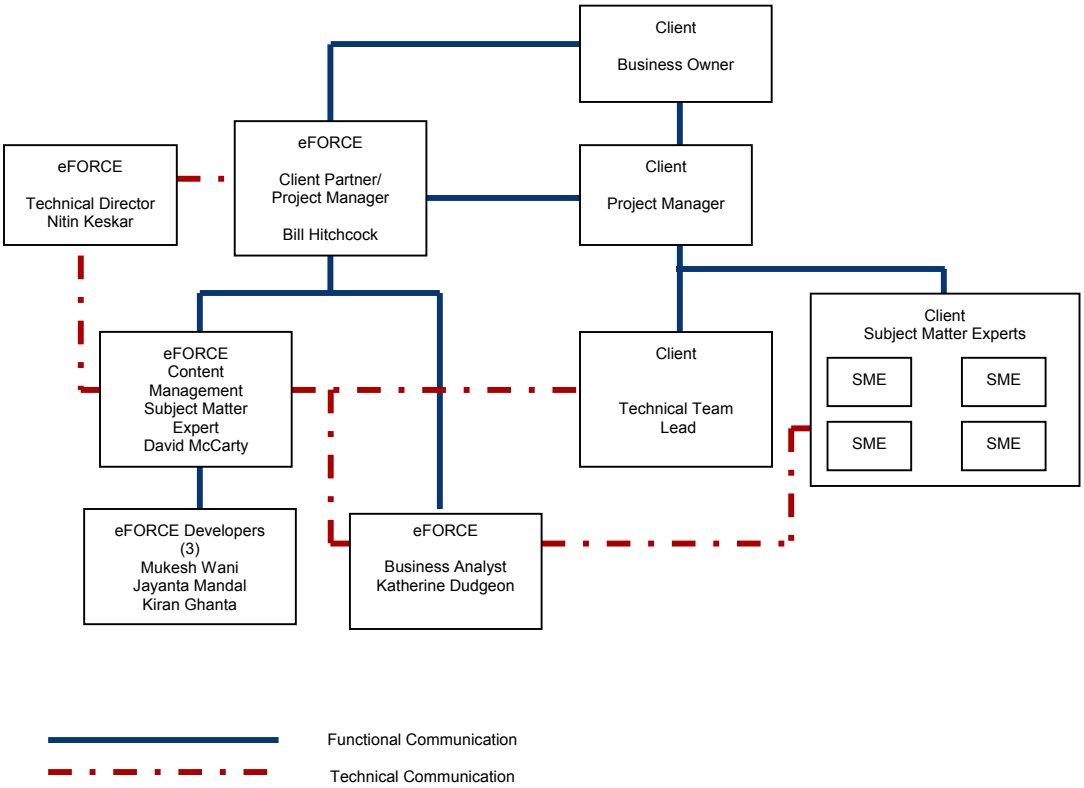


5.8. Communication Plan

A comprehensive project plan is composed of many components. While schedule and budget generally receive the most attention, it is the Communications Plan that is, consciously or unconsciously, employed most often. Without documented and adhered to channels of communication and accountability, projects become mired in confusion and disarray, with little chance of success.

eFORCE strives to facilitate communication through construction of a Communications Matrix. This graphical tool illustrates both functional (accountability) and technical (sharing of knowledge) lines of communication.

The Communications Matrix also serves as an escalation and issues resolution path for all project participants. It is very important that channels of communication are designed for facilitation rather than restriction. A prototype Communications Plan is provided below for illustration.



In addition to the Communications Matrix, separate Communications Plan sub-plans or reference documents are constructed covering:

- Weekly Status Reports
- Regularly scheduled meetings
- Periodic reports
- Stakeholder and team contact lists
- Escalation policies
- Change Control (Also part of Scope Planning)
- Issues Logs
- Knowledge Transfer Plan

An additional component of the Communications Plan, which deserves special emphasis, is the Knowledge Transfer Plan. Too often, consulting engagements end with a client holding only program code and arcane technical manuals. eFORCE believes that transfer of technical knowledge is as important as delivery of the system itself. Knowledge Transfer is an interactive process, concluding when the client is satisfied that their personnel are fully prepared to assume responsibility for operation and maintenance.

The Knowledge Transfer Plan begins with identifying technical deliverables requiring transfer of specialized knowledge. Documentation standards are defined as part of Requirements definition during the Inception Phase. Specific personnel

on the client side are identified as having responsibility for receiving knowledge transfer. As each technical deliverable is met, a Knowledge Transfer Statement is signed by both eFORCE and client technical resources.

No technical deliverable is considered final until the Knowledge Transfer Statement is executed. This assures that eFORCE and the client are satisfied that knowledge of the system, as well as the system itself, has been delivered.

Each sub-plan incorporates (as appropriate) member lists, distribution lists, timetables and ownership assignment.

5.9. Success Criteria

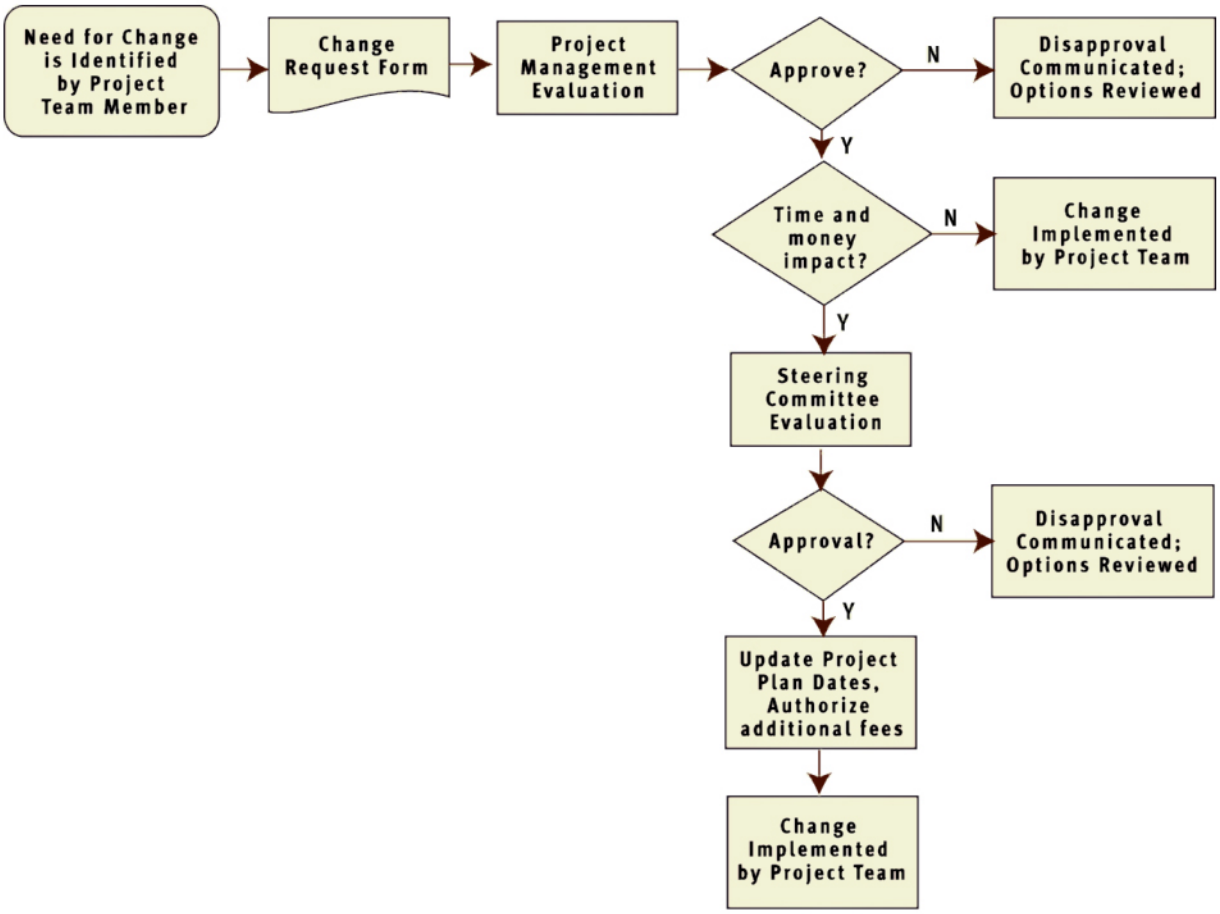
It is also imperative that the teams jointly define success criteria that can serve as important metrics that the entire team can rally toward throughout the entire project.

In addition, it is imperative that the client and eFORCE have a shared view on Acceptance Criteria that are based on parameters relevant to the application—Functionality, Performance, Usability, etc. In the case of Fixed Price contracts, this Acceptance Document serves as the unambiguous criteria for transition of the solution to the client for post-deployment support.

6. Change Control

The purpose of the Change Control process is to provide the standards and procedures for submitting, analyzing and approving or disapproving changes that occur after the official requirements have been agreed upon (i.e., "signed off"). This process provides a framework for minimizing changes that impact the scope, schedule, and cost of the project, while facilitating project team members' ability to make simple improvements to the solution.

Changes in the project approach, business requirements, technology, and resources will impact key components of the implementation plan. Such changes must be managed and controlled by the project management team, in order to prevent increased customer cost, delayed delivery schedule, and decreased quality of the overall solution. The process also provides a method for tracking decisions, evaluating impacts, and communicating changes.



6.1. Change Control Process

6.1.1. Change Control Process Overview

The following describes the Change Control process (as outlined in the diagram above):

STEP 1: A project team member identifies the need for a change in an area where the requirements, approach, or plan has already been approved (i.e., "signed off") by the project management team. He/she (i.e., "Change Originator") completes a 'Change Request Form' and forwards it to his/her project manager—either the Client Project Manager or eFORCE Project Manager. The completed request describes the business, technical, or change management issue the request is trying to address, the desired outcome of the change, and detailed justification for the change.

STEP 2: The project manager analyzes the initial change request. If it is valid, he/she submits the request to the eFORCE Project Manager, who logs the request in the Change Control Log. The eFORCE Project Manager evaluates the priority of the request and arranges a meeting (via PM Review, Issue Management, or ad-hoc meeting) for the entire project management team to discuss and analyze.

STEP 3: The entire project management team analyzes the request and agrees to approve or disapprove. If the request is disapproved, the originating team member's project manager communicates the outcome, and reviews options/workarounds (as discussed in the project management review of the request). The eFORCE Project Manager logs the request as disapproved on the Change Control log.

If the request is approved, the project management must thoroughly evaluate any impacts to the project scope, schedule, and cost. If there are any impacts to these stated areas, the request must be presented to the Steering Committee for approval. If there are no impacts to project scope, schedule, or cost, the change is communicated to the originating team by their respective project manager. The eFORCE Project Manager logs the request as approved on the Change Control Log, and communicates it on the weekly status report.

STEP 4: If the request is approved, and there is an impact to scope, schedule, or cost, the eFORCE Project Manager arranges for a Steering Committee review. The Steering Committee evaluates the request and makes a decision to approve or disapprove. If the decision is disapproval, the originating team's project manager communicates the outcome to his/her respective team members, and reviews options/workarounds (as discussed with the Steering Committee members). The eFORCE Project Manager logs the request as disapproved on the Change Control Log.

If the request is approved, the respective project manager communicates the change to their originating team. The eFORCE Project Manager logs the request as approved on the Change Control Log and communicates on the weekly status report. The project plan is revised to reflect the change in scope and delivery dates, and the client authorizes the additional funding as necessary, often via a purchase order.

6.2. Change Analysis Framework

Listed below are the criteria that should be used to determine whether or not the change request should be approved or disapproved:

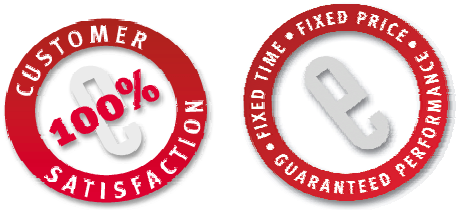
- Does the change include valid justification?
- Is the change essential at this point in the project, or can it be deferred to post-production efforts?
- Is the change outside the agreed-upon scope of this stage of the project?

- What tasks will be impacted by this change? Are there any critical path tasks?
- What is the work effort estimated to implement this change?
- What is the impact to the project schedule?
- Are additional resources required to implement this change?
- Are any other applications or projects impacted by this change?
- Are there any risks associated with implementing this change?
- Are there any costs associated with this change?

7. SUMMARY

In summary:

- eFORCE believes that a methodology in itself cannot assure project success. Furthermore, a proprietary or highly rigid/inflexible methodology can lead to project failure.
- One needs to address all the critical success factors—People, Product/Technology, and Place/Environment—in addition to the Process/Methodology aspects.
- For the basis of Process/Methodology, it is imperative to use open, published and de-facto industry standards.
- eFORCE's project approach leverages the de-facto standard methodology, RUP, with best practices and approaches derived from over 200 successful (on-time, on-budget, high-quality) engagements at Global 1000 companies.



8. REFERENCES

White Papers/Articles

- What Is the Rational Unified Process? - http://www.therationaledge.com/content/jan_01/f_rup_pk.html
- The Ten Essentials of RUP - http://www.therationaledge.com/content/dec_00/f_rup.html
- Rational Unified Process for Systems Engineering - http://www.rational.com/media/whitepapers/TP165.pdf
- Developing Large Scale Systems Using RUP - http://www.rational.com/media/whitepapers/sis.pdf
- New Dimensions of Project Management - http://www.therationaledge.com/content/may_01/f_projman_sf.html
- Software's Ten Essentials - IEEE Software, Best Practices, Vol. 14, No. 2, March/April 1997, http://www.construx.com/stevemcc/ieeesoftware/bp08.htm

Vendor Documentation

- Rational Unified Process - Process Made Practical - http://www.rational.com/products/rup/poster.jsp#

Industry Associations

- Project Management Institute - http://www.pmi.org
- Carnegie Mellon Software Engineering Institute - http://www.sei.cmu.edu/

Books

- Software Project Survival Guide: How to Be Sure Your First Important Project Isn't Your Last, Steve McConnell, 1997.
- Rapid Development: Taming Wild Software Schedules Steve McConnell, 1997.
- Managing the Software Process, Watts S. Humphrey, 1996