Tutorial | Leverage the Software Development Lifecycle
IT METHODOLOGY WEBINAR
Things to know...

- All participants will be on mute
- Questions are welcome
- Ask questions in the question box
- We DO NOT send out PDU certificates
- May be valid for PDU credit (self-report session)
- Recording and slides sent by tomorrow
Moderator

Denise Rodriguez
Project Insight
Marketing
Denise.Rodriguez@projectinsight.com
www.projectinsight.net
Schedule a customized demo today!

- +1 (949) 476-6499 x3
- info@projectinsight.net
- Request info: www.projectinsight.net
Cameron Watson
President, QAAssistant
cwatson@qaassist.com
www.qaassist.com
Who’s on the call?

- I am a Project Manager
- I am a Business or Operational Manager
- I am a software developer or software tester
- I am a member of an organizational PMO, governance or quality assurance function
- I want a better understanding of IT Methodology
Who is QAIassist?

- Established 2007 – Ottawa, Canada
- Nurturing organizational efficiency through IT Methodology knowledge & expertise
- Support and promote a community with evolving IT Methodology knowledge and expertise
- Maintain, optimize, promote and support the delivery of IT methodology products and lifecycles proven to increase operational performance
- Administration and oversight of QAIassist Integrated Methodology (QAIassist-IM) and formal certifications – “Foundation” & “Practitioner”
- Support practitioners and authorized affiliates in acquiring and delivering IT Methodology knowledge and expertise
Nurturing efficiency through IT Methodology knowledge & expertise across the globe

- Over 200 countries
- Wide array of industries (manufacturing, banking, health, insurance, consulting)
- More than 50 unique webinars

Over 10,000 webinar registrants

- Executives
- Operational & Delivery Managers
- Project Managers
- Business Analysts and Architects
- Application Developers & Testers
- Corporate PMO, Quality Assurance, Continuous Improvement
- Consultants
QAlassist 2017 Series - Goals

- Share IT Methodology knowledge
- Develop IT methodology understanding & expertise
- Provide additional insight into several IT (project management, software development, software testing) methodologies
- Discuss best practices on incorporating an organizational IT methodology
- Promote ongoing dialog and feedback with webinar audience (questions, email, survey)
2017 Schedule

- The 3rd Wednesday of Every Month
  - 8:00 am Pacific - 11:00 am Eastern Time

- IT Methodology Concepts (60 mins)
  - This series of six webinars will be presented on a bi-monthly basis. Each webinar will provide you with a context, overview, terminology and general understanding of information technology (IT) methodology.

- IT Methodology – Tutorials (60 mins)
  - This series of five webinars will be presented on a bi-monthly basis. Each session is designed to offer a more “hands on” perspective – bridging the concepts into a practical utilization.
2017 Concept Sessions

The 3rd Wednesday of Month (Jan, Mar, May, Jul, Sept, Nov)

- 8:00 am Pacific - 11:00 am Eastern Time

IT Methodology Concepts (60 mins)

- A Context for IT Methodology - (Jan)
- Apply PM Fundamentals to IT - (Mar)
- Leverage the Software Development Lifecycle - (May)
- Explore the Software Testing Lifecycle - (Jul)
- Incorporate Organizational Process Governance - (Sept)
- Implementing IT Methodology - (Nov)
2017 Tutorial Sessions

- The 3rd Wednesday of Month (Feb, Apr, Jun, Aug, Oct)
  - 8:00 am Pacific - 11:00 am Eastern Time

- IT Methodology Tutorials (60 mins)
  - Applying IT Methodology to Project Initiation - (Feb)
  - Project Planning & Design with IT in Mind - (Apr)
  - Executing Projects with IT Methodologies - (Jun)
  - Project Control & Verification - (Aug)
  - Project Close & Delivery - (Oct)
2017 Concept Sessions

- The 3rd Wednesday of Month (Jan, Mar, May, Jul, Sept, Nov)
  - 8:00 am Pacific - 11:00 am Eastern Time

- IT Methodology Concepts (60 mins)
  - A Context for IT Methodology - (Jan)
  - Apply PM Fundamentals to IT - (Mar)
  - Leverage the Software Development Lifecycle - (May)
  - Explore the Software Testing Lifecycle - (Jul)
  - Incorporate Organizational Process Governance - (Sept)
  - Implementing IT Methodology - (Nov)
2017 Concept Sessions

The 3rd Wednesday of Month (Jan, Mar, May, Jul, Sept, Nov)
- 8:00 am Pacific - 11:00 am Eastern Time

IT Methodology Concepts (60 mins)
- A Context for IT Methodology - (Jan)
- Apply PM Fundamentals to IT - (Mar)
- Leverage the Software Development Lifecycle - (May)
- Explore the Software Testing Lifecycle - (Jul)
- Incorporate Organizational Process Governance - (Sept)
- Implementing IT Methodology - (Nov)
Webinar Goals

- Review previous IT Methodology Concepts Sessions (Jan, Mar)
- Define a context for Software Development Lifecycle (SDLC)
- Discuss significance of Requirements and Traceability
- Identify the purpose and objectives of SDLC phases
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- SDLC Phases - Objectives, Iterations, Traceability
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- SDLC Phases - Objectives, Iterations, Traceability
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
Terms, Terms and more Terms

IT Framework, IT Methodology, IT Lifecycle

- Non Standard usage in terminology
- Terms are used interchangeably between organizations and across the IT industry
- Examples include
  - Project Management Methodology/Lifecycle/Framework
  - Software Development Methodology/Lifecycle (SDLC)/Framework
  - Software Testing Methodology/Lifecycle/Framework
- Every IT organization is using an IT Methodology or IT Lifecycle or IT Framework
IT Methodology Context

- **Organizational Frameworks**
  - ISO(900XXX), CMMI, Six-Sigma, ITIL, COBIT

- **Methodologies/Lifecycles (noun)**
  - Rational Unified Process (RUP), QAllassist (IM), PMI, Prince2

- **Methodologies/Lifecycles (verb)**
  - Waterfall, Spiral, Agile, RAD, etc
IT Methodology Context

- **Organizational Frameworks**
  - ISO(900XXX), CMMI, Six-Sigma, ITIL, COBIT

- **Methodologies/Lifecycles (noun)**
  - Rational Unified Process (RUP), QAassist (IM), PMI, Prince2

- **Methodologies/Lifecycles (verb)**
  - Waterfall, Spiral, Agile, RAD, etc
IT Methodology – Map & Journey

- Methodologies/Lifecycles - “noun”
  - The roadmap
  - Rational Unified Process (RUP), QAassist (IM), PMI, Prince2

- Methodologies/Lifecycles - “verb”
  - The journey taken along the road
  - Waterfall, Spiral, Agile, RAD, Prototyping, etc
All Methodologies/Lifecycles (RUP, QAIassist-IM, PMI, Prince2) define, utilize and rely on their own pre-defined structure, naming conventions, and terminology.

Hierarchy/Structure

- Methodology/Lifecycle (ie Project Management, Software Development, Software Testing)
- Phase/Stage (ie Initiate, Planning, Design, Closeout)
- Deliverable/Artifact/Work Product (ie Project Charter, Testing Strategy)
- Activity (tasks performed to complete deliverables, artifacts, work products)
# QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Team Status Report</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Project Plan</td>
<td>Project Status Report</td>
<td></td>
</tr>
<tr>
<td>Project Schedule/WBS</td>
<td>Project Risk Definition</td>
<td>Unit Test (UT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Issue Log</td>
<td>System Integration Test (SIT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Project Change Request Definition</td>
<td>User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Config. Management Plan</td>
<td>Project Change Request Log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Procedures</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Systems Analysis

<table>
<thead>
<tr>
<th>Software Development</th>
<th>Entry Deliverables</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>High Level Solution Design</td>
<td>Unit Code</td>
<td>Exit Deliverables</td>
<td>Unit Code</td>
</tr>
<tr>
<td>Detailed Business Requirements</td>
<td>Detailed Solution Design</td>
<td>Regmt's Traceability Log</td>
<td>Regmt's Traceability Log</td>
<td>Regmt's Traceability Log</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>Programming Specifications</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>High Level Solution Design</td>
<td>Regmt's Traceability Log</td>
<td>Unit Code</td>
<td>Unit Code</td>
<td>Unit Code</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>Training &amp; Support Plan</td>
<td>(UT) Plan</td>
<td>(UT) Plan</td>
<td>(UT) Plan</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>(SIT) Evaluation Criteria</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>High Level Solution Design</td>
<td>(SIT) Plan</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>Testing Strategy</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Detailed Solution Design</td>
<td>(SIT) Plan</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>(SIT) Evaluation Criteria</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>High Level Solution Design</td>
<td>(SIT) Plan</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>(SIT) Evaluation Criteria</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Detailed Solution Design</td>
<td>(SIT) Plan</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>(SIT) Evaluation Criteria</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>High Level Solution Design</td>
<td>(SIT) Plan</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>(SIT) Evaluation Criteria</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Detailed Solution Design</td>
<td>(SIT) Plan</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>(SIT) Evaluation Criteria</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>High Level Solution Design</td>
<td>(SIT) Plan</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>(SIT) Evaluation Criteria</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
</tbody>
</table>
Building an IT Application

Deliverable (Artifact, Work Product)
- Business Case
- Project Plan
- User Acceptance Test Plan

Activity (non-Deliverable, non-Artifact, non Work Product)
- Reviewing business need with Operational Management, Business Analysts and End-Users
- Meeting with Stakeholders to acquire approval of Project Plan
- Discussing User Acceptance Testing objectives and priorities with testing team
Fixed Sections (always populated)

- Title Page
- Deliverable Details (Author, Creation Date, Version, Status, etc)
- Deliverable History, Reviewers, Sign Off
- Table of Contents (as per deliverable)
- Context (purpose of the deliverable)

Variable Sections (based on pre-defined informational requirements)

- Structure/format defined as per deliverable (i.e., a Project Charter deliverable does not have the same pre-defined informational requirements as a Testing Strategy deliverable)
Draft - Deliverable has been identified, assigned and is in the process of being drafted/completed

Review - “Draft” deliverable has been completed and submitted for “review” and “authorization”

Approved/Archived - “Reviewed” deliverable has been “Authorized” by designated stakeholders and put under configuration management

Applied - Information within “Authorized” deliverable is being referenced and/or utilized within another deliverable
Tangible (pre-defined informational requirements)

- Can be identified and defined (completion dates, cost) on Project Schedule
- Can be assigned (populating information requirements) to project team member(s)
- Can be referenced by project team members to support creation of other deliverables
- Can be formally reviewed and officially authorized
- Will alter in status (authored, reviewed, approved, archived, referenced) throughout life of project
- Will reflect evolving deliverable and project status/progress (risks, issues, percentage complete)
- Can be placed under formal configuration management – re: with evolving status
- Can be accessed for future reference (application support and maintenance)
Viable IT methodologies (noun and verb) are designed to help not hinder

IT methodologies (noun) are applicable for a wide range of IT Methodologies (verb) and delivery approaches (waterfall, spiral, agile, RAD, prototyping, etc)

No hard or specific rules (i.e., iterations, frequency of iterations, deliverables to be completed) of how an IT methodology (noun) can best be applied or leveraged by an organization or project team.

No silver bullet - if the perfect combination of IT Methodology (noun) and IT Methodology (verb) existed, every project team and organization would be using it.
Deliverables (and content) may be re-usable from one project to another.

- Not every deliverable of every phase has to be completed for every project - any and every deliverable to be completed must add value (cost, schedule, quality) to the project.

- Not every section or sub-section of every deliverable has to be completed for every project - information used to populate every deliverable must add value (cost, schedule, quality) to the project.
# QAlassist Integrated Methodology

## Initiative
- Business Case
  - Detailed Business Req’s
- Project Charter
  - Project Name
- Project Schedule/WS:
  - Roles & Responsibilities
  - Project Deliverables
  - Config, Management Plan
  - Quality Assurance Plan
  - Project Procedures

## Plan
- All previously created PM deliverables
  - Project Risk Definition
  - Project Plan
  - Project Issue Logs
  - Project Change Request Definition
  - Project Change Request Log

## Execute & Control
- Team Status Report
  - Project Status Report
  - Unit Test (UT) Authorization
  - System Integration Test (SIT) Authorization
  - User Acceptance Test (UAT) Authorization

## Closeout
- Project Closeout

## Systems Analysis
- Entry Deliverables
  - Business Case
    - Program Charter
    - Detailed Business Requirement
  - High Level Solution Design
    - Req’ts Traceability Log

## Design
- Entry Deliverables
  - High Level Solution Design
    - Req’ts Traceability Log
  - Detailed Solution Design
    - Programming Specifications
      - Req’ts Traceability Log

## Build
- Entry Deliverables
  - Detailed Solution Design
    - Programming Specifications
      - Req’ts Traceability Log
  - Training & Support Plan
    - Req’ts Traceability Log

## Test
- Entry Deliverables
  - Unit Code
    - (UT) Evaluation Criteria
      - Req’ts Traceability Log
  - (UT) Defect Log
    - Req’ts Traceability Log

## Release
- Entry Deliverables
  - Unit Code
    - (UT) Authorization
    - (SIT) Authorization
    - (UAT) Authorization
  - (UT) Evaluation Criteria
    - (SIT) Evaluation Criteria
    - (UAT) Evaluation Criteria
  - (UT) Defect Log
    - (SIT) Defect Log
    - (UAT) Defect Log
  - Req’ts Traceability Log
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- SDLC Phases - Objectives, Iterations, Traceability
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT

Software Development Lifecycle (SDLC) – Context and Premise

SDLC - Requirements (Identification and Definition)

SDLC Phases - Objectives, Iterations, Traceability

SDLC – Internal Dynamics (PMLC and STLC interfaces)

Recap
SDLC – Business Context

**Development Environment**
- New Products
- New Operational Business Applications (ie HR, Accounting, Sales, etc)

**Maintenance/Support Environment**
- Existing Products
- Existing Operational Business Applications (HR, Accounting, Sales, etc)
- Off the Shelf Applications
Multiple IT Environments
- Mainframe, Distributed, Web Based

Multiple Delivery Approaches – Methodology (verb)
- Waterfall
- Agile
- Prototyping

Integrations
- Project Management Lifecycle
- Software Testing Lifecycle
How many requirements (and associated functionality) should/can be developed?

- Too Few – leads to a solution that does not satisfy the business need
- Too Many – leads to a solution that is over priced, over built, over budget and past due (and may still not satisfy the business need)

How can a project team know when they are best leveraging an SDLC to develop the appropriate level of functionality?
Identifying, Designing, Building and Unit Testing functionality

**** Functionality cannot be delivered in a vacuum ****

- **Manual Functionality**
  - Processes that deliver or administer the delivery of products and/or services (often leverages automated functionality)

- **Automated Functionality**
  - IT Systems/Applications that interface and support the delivery of Manual Functionality
SDLC – Premise
SDLC – Premise
SDLC – Premise

Project

Process Model

Process

Process

Process

Functionality

Application Programs

Data Model

Data

Data

Data

Logical Model

Database

I/O Structures
SDLC – Premise
Generic Cohesion

- “Data” and “Process” exist – a correlation exists between them
- “Data” identifies and defines the information that will be utilized by the application to satisfy the business need
- “Process” defines how the “Data” will be utilized/leveraged to satisfy the business need

Data

- “Subject” – highest level definition of user requirement - correlation with “System”
- “Topics” – breakdown of “Subject” - correlation with “Sub-Systems”
- “Entities” - breakdown of “Topics” - correlation with “Functions”

Process

- “System” – highest level “process” – correlation with “Subject”
- “Sub-System(s)” – breakdown of “System” – correlation with “Topics”
- “Function(s)” – breakdown of “Sub-Systems” – correlation with “Entities”
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- **Software Development Lifecycle (SDLC) – Context and Premise**
- SDLC - Requirements (Identification and Definition)
- SDLC Phases - Objectives, Iterations, Traceability
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- SDLC Phases - Objectives, Iterations, Traceability
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
SDLC – Requirements (Identification and Definition)
Sample Application
SDLC – Requirements (Identification & Definition)
“Data” Context

Subject

Topics

Entities

Accounting

Receivables
Customer Attained
Services Calculated
Invoice Sent
Payment Received

Payables
Invoice Received
Invoice Paid

Payroll
Hours Worked
Vacation Days
Wages Paid
SDLC – Requirements (Identification & Definition) 
“Process” Context

Accounting
- Receivables
  - Customer Attained
  - Services Calculated
  - Invoice Sent
  - Payment Received
- Payables
  - Invoice Received
  - Invoice Paid
- Payroll
  - Hours Worked
  - Vacation Days
  - Wages Paid
SDLC – Requirements (Identification & Definition) "Process" Context

- Accounting
  - Receivables
    - Customer Attained
    - Services Calculated
    - Invoice Sent
    - Payment Received
  - Payables
    - Invoice Received
    - Invoice Paid
  - Payroll
    - Hours Worked
    - Vacation Days
    - Wages Paid

- System
- Sub-Systems
- Functions
SDLC – Requirements (Identification & Definition)
“Process” Context

- System
- Sub-Systems
- Functions
- Modules/Programs

**Accounting**
- Receivables
  - Customer Attained
  - Services Calculated
  - Invoice Sent
  - Payment Received
- Payables
  - Invoice Received
  - Invoice Paid
- Payroll
  - Hours Worked
  - Vacation Days
  - Wages Paid
- Client Marketing
- Sales Promotion
- CRM
SDLC – Requirements (Identification & Definition)
“Process” – Naming Convention

- System
  - Accounting: S1
  - Payables: SS2
  - Payroll: SS3

- Sub-Systems
  - Receivables: SS1
  - Payables: SS2
  - Payroll: SS3

- Functions
  - Customer Attained: SS1-F1
  - Services Calculated: SS1-F2
  - Invoice Sent: SS1-F3
  - Payment Received: SS1-F4
  - Invoice Received: SS2-F1
  - Invoice Paid: SS2-F2
  - Hours Worked: SS3-F1
  - Vacation Days: SS3-F2
  - Wages Paid: SS3-F3

- Modules/Programs
  - Client Marketing: SS1-F1-P1
  - Sales Promotion: SS1-F1-P2
  - CRM: SS1-F1-P3

**Diagram:**
- Accounting
- Payables
- Payroll
- Receivables
- Payables
- Payroll
- Customer Attained
- Services Calculated
- Invoice Sent
- Payment Received
- Invoice Received
- Invoice Paid
- Hours Worked
- Vacation Days
- Wages Paid
- Client Marketing
- Sales Promotion
- CRM
Generic Cohesion

- “Data” and “Process” exist – a correlation exists between them
- “Data” identifies and defines the information that will be utilized by the application to satisfy the business need
- “Process” defines how the “Data” will be utilized/leveraged to satisfy the business need

Data

- “Subject” – highest level definition of user requirement - correlation with “System”
- “Topics” – breakdown of “Subject” - correlation with “Sub-Systems”
- “Entities” - breakdown of “Topics” - correlation with “Functions”

Process

- “System” – highest level “process” – correlation with “Subject”
- “Sub-System(s)” – breakdown of “System” – correlation with “Topics”
- “Function(s)” – breakdown of “Sub-Systems” – correlation with “Entities”
<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Team Status Report</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Project Plan</td>
<td>Project Status Report</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project Team WBS</td>
<td>Unit Test (UT) Authorization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Roles &amp; Responsibilities</td>
<td>System Integration Test (SIT) Authorization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project Deliverables</td>
<td>User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Config, Management Plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quality Assurance Plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project Procedures</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Systems Analysis**

<table>
<thead>
<tr>
<th>Software Development</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Deliverables</td>
<td>Entry Deliverables</td>
<td>Entry Deliverables</td>
<td>Entry Deliverables</td>
<td>Entry Deliverables</td>
</tr>
<tr>
<td>Business Case</td>
<td>High Level Solution Design</td>
<td>Detailed Solution Design</td>
<td>Unit Code</td>
<td>Unit Code</td>
</tr>
<tr>
<td>Project Charter</td>
<td>Req'ts Traceability Log</td>
<td>Req'ts Traceability Log</td>
<td>(UT) Plan</td>
<td>(UT) Plan</td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Programming Specifications</td>
<td>Req'ts Traceability Log</td>
<td>Req'ts Traceability Log</td>
<td>Req'ts Traceability Log</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>High Level Solution Design</td>
<td>Detailed Solution Design</td>
<td>Training &amp; Support Plan</td>
<td>Unit Code</td>
<td>Unit Code</td>
</tr>
</tbody>
</table>

**Software Testing**

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Entry Deliverables</th>
<th>Entry Deliverables</th>
<th>Entry Deliverables</th>
<th>Entry Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing Strategy</td>
<td>High Level Solution Design</td>
<td>Detailed Solution Design</td>
<td>Testing Strategy</td>
<td>(UAT) Plan</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
</tbody>
</table>

QAassist Integrated Methodology
<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>All previously created PM deliverables</td>
<td>Team Status Report</td>
</tr>
<tr>
<td>Detailed Business Req’s</td>
<td>Project Plan</td>
<td>Project Risk Definition</td>
<td>Project Status Report</td>
</tr>
<tr>
<td>Project Schedule/WBS</td>
<td>Roles &amp; Responsibilities</td>
<td>Project Issue Log</td>
<td>Unit Test (UT) Authorization</td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Config. Management Plan</td>
<td>Project Change Request Definition</td>
<td>System Integration Test (SIT) Authorization</td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td>Project Procedures</td>
<td>Project Change Request Log</td>
<td>User Acceptance Test (UAT) Authorization</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Systems Analysis**

<table>
<thead>
<tr>
<th>Software Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry Deliverables</td>
</tr>
<tr>
<td>Detailed Business Requirement</td>
</tr>
<tr>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Reqmts’ Traceability Log</td>
</tr>
</tbody>
</table>

**Design**

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>High Level Solution Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reqmts’ Traceability Log</td>
<td></td>
</tr>
</tbody>
</table>

**Build**

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Detailed Solution Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Specifications</td>
<td></td>
</tr>
<tr>
<td>Reqmts’ Traceability Log</td>
<td></td>
</tr>
</tbody>
</table>

**Test**

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Unit Code (UAT Plan)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(UT) Evaluation Criteria</td>
<td></td>
</tr>
<tr>
<td>Reqmts’ Traceability Log</td>
<td></td>
</tr>
</tbody>
</table>

**Release**

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Unit Code (UT) Authorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SIT) Plan</td>
<td></td>
</tr>
<tr>
<td>(SIT) Evaluation Criteria</td>
<td></td>
</tr>
<tr>
<td>Reqmts’ Traceability Log</td>
<td></td>
</tr>
</tbody>
</table>
SDLC - Mechanics
High Level Solution Design Deliverable (S1, SSX, SSX-FX)
## QAlassist Integrated Methodology

### Initiate
- Business Case
- Detailed Business Requirements
- Project Charter

### Plan
- Project Plan
- Project Schedule/WBS
- Roles & Responsibilities
- Project Deliverables
- Config. Management Plan
- Quality Assurance Plan
- Project Procedures

### Execute & Control
- All previously created PM deliverables
- Project Risk Definition
- Project Risk Log
- Project Issue Definition
- Project Issue Log
- Project Change Request Definition
- Project Change Request Log

### Closeout
- Project Closeout
- Team Status Report
- Project Status Report
- Unit Test (UT) Authorization
- System Integration Test (SIT) Authorization
- User Acceptance Test (UAT) Authorization

### Systems Analysis
- **Entry Deliverables**
  - Business Case
  - Project Charter
  - Detailed Business Requirements

- **Exit Deliverables**
  - High Level Solution Design
  - Regmt’s Traceability Log

### Design
- **Entry Deliverables**
  - High Level Solution Design
  - Programming Specifications

- **Exit Deliverables**
  - Detailed Solution Design
  - Training & Support Plan
  - Regmt’s Traceability Log

### Build
- **Entry Deliverables**
  - Detailed Solution Design
  - Programming Specifications

- **Exit Deliverables**
  - Unit Code
  - (UT) Evaluation Criteria

### Test
- **Entry Deliverables**
  - Unit Code
  - (UT) Plan

- **Exit Deliverables**
  - Unit Code
  - (UT) Evaluation Criteria

### Release
- **Entry Deliverables**
  - Unit Code
  - (UT) Authorization

- **Exit Deliverables**
  - Unit Code
  - (UT) Authorization

---

QAlassist 2.1
SDLC - Mechanics

Detail Solution Design Deliverables (SSX-FX one per)

- **Accounting**
  - **SS1**
    - **Receivables**
      - **SS1**
        - Customer Attained (SS1-F1)
        - Services Calculated (SS1-F2)
        - Invoice Sent (SS1-F3)
        - Payment Received (SS1-F4)
    - **Payables**
      - **SS2**
        - Invoice Received (SS2-F1)
        - Invoice Paid (SS2-F2)
    - **Payroll**
      - **SS3**
        - Hours Worked (SS3-F1)
        - Vacation Days (SS3-F2)
        - Wages Paid (SS3-F2)
SDLC - Mechanics
Programming Specifications (SSX-FX-PX one per)

- Accounting
  - Receivables SS1
  - Customer Attained SS1-F1
    - Services Calculated SS1-F1
    - Invoice Sent SS1-F1
    - Payment Received SS1-F1
  - Payment Sent
  - Payment Received

- System
  - Sub-Systems
    - Functions
      - Modules/Programs
        - Client Marketing SS1-F1-P1
        - Sales Promotion SS1-F1-P2
        - CRM SS1-F1-P3
Software Development
Requirements Deliverables & Dependencies

- **High Level Solution Design deliverable**
  - Identifies data “Subject” and process “Systems”
  - Identifies data “Topics” and process “Subsystems”
  - Identifies data “Entities” and process “Functions”

- **Detailed Solution Design deliverable(s)**
  - One deliverable for every “Function” as identified in the High Level Solution Design deliverable
  - Functionality defined (Manual and Automated)
  - All required “Unit Modules/Programs” are identified

- **Programming Specification deliverable(s)**
  - One deliverable for every “Unit Module/Program” as identified in the Detailed Solution Design deliverables
  - Specifics for each “Unit Module/Program” are described
Establish webinar context - review previous “Concepts” sessions

- Jan - A Context for IT Methodology
- Mar - Apply PM Fundamentals to IT

Software Development Lifecycle (SDLC) – Context and Premise

SDLC - Requirements (Identification and Definition)

SDLC Phases - Objectives, Iterations, Traceability

SDLC – Internal Dynamics (PMLC and STLC interfaces)

Recap
Establish webinar context - review previous “Concepts” sessions

- Jan - A Context for IT Methodology
- Mar - Apply PM Fundamentals to IT

Software Development Lifecycle (SDLC) – Context and Premise

SDLC - Requirements (Identification and Definition)

SDLC Phases - Objectives, Iterations, Traceability

SDLC – Internal Dynamics (PMLC and STLC interfaces)

Recap
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concept” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- SDLC Phases - Objectives, Iterations, Traceability
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
Software Development Lifecycle (SDLC) Objectives

- Establish a foundation for ongoing collaboration and communication between Business/User Community and Information Technology/IT resources
- Business/User community contributes to and authorizes formal requirements and deliverables throughout the completion of the project – necessary documentation
- Computer **Unit Code** is developed/built to reflect the authorized and required functionality (design and business requirements)
- Computer **Unit Code** is successfully **unit tested** against the authorized design and business requirements - satisfactory code is made available for additional testing
- End User training is planned and delivered
SDLC objectives (see previous page) are achieved utilizing five phases, they are:

- Systems Analysis
- Design
- Build
- Test (Unit)
- Release
## IT Methodology

### SDLC

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>All previously created PM deliverables</td>
<td>Team Status Report</td>
</tr>
<tr>
<td>Detailed Business Requirements</td>
<td>Project Plan</td>
<td>Project Risk Definition</td>
<td>Project Status Report</td>
</tr>
<tr>
<td>Project WBS</td>
<td>Roles &amp; Responsibilities</td>
<td>Project Issue Log</td>
<td>Unit Test (UT) Authorization</td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Project Change Request Definition</td>
<td>Project Change Request Log</td>
<td>System Integration Test (SIT) Authorization</td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td>Project Procedures</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Systems Analysis

#### Entry Deliverables
- Business Case
- Detailed Business Requirements

#### Design

#### Build

#### Test

#### Release

### Software Development

#### Entry Deliverables
- High Level Solution Design
- Detailed Solution Design
- Unit Code
- UAT Plan

#### Exit Deliverables
- High Level Solution Design
- Detailed Solution Design
- Unit Code
- UAT Plan

### Software Testing

#### Entry Deliverables
- Test Strategy
- SIT Plan (UAT) Evaluation Criteria
- Exit Deliverables
- SIT Plan (UAT) Evaluation Criteria

#### Exit Deliverables
- Test Strategy
- SIT Plan (UAT) Evaluation Criteria

### QAassist Integrated Methodology

#### QAassist 2.1
<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Team Status Report</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Project Plan</td>
<td>Project Status Report</td>
<td></td>
</tr>
<tr>
<td>SDLC</td>
<td>Project Schedule/WBS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Risk Log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Project Issue Log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Config, Management Plan</td>
<td>Project Change Request Definition</td>
<td>System Integration Test (SIT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td>Project Change Request Log</td>
<td>User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Project Procedures</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Systems Analysis**

- **Entry Deliverables**
  - Business Case
  - Detailed Business Requirements
- **Key Deliverables**
  - High Level Solution Design
  - Regent’s Traceability Log
- **Exit Deliverables**
  - High Level Solution Design
  - Regent’s Traceability Log

**Design**

- **Entry Deliverables**
  - High Level Solution Design
  - Testing Strategy
- **Key Deliverables**
  - Detailed Solution Design
  - Programming Specifications
  - Regent’s Traceability Log
- **Exit Deliverables**
  - Detailed Solution Design
  - Programming Specifications
  - Regent’s Traceability Log

**Build**

- **Entry Deliverables**
  - Unit Code
  - Unit (UT) Plan
  - Unit (UT) Plan (UAT) Evaluation Criteria
- **Key Deliverables**
  - Unit Code
  - Training & Support Plan
  - Regent’s Traceability Log
- **Exit Deliverables**
  - Unit Code
  - Training & Support Plan
  - Regent’s Traceability Log

**Test**

- **Entry Deliverables**
  - Unit Code
  - (UAT) Evaluation Criteria
  - Regent’s Traceability Log
- **Key Deliverables**
  - Unit Code
  - (UT) Evaluation Criteria
  - Regent’s Traceability Log
- **Exit Deliverables**
  - Unit Code
  - (UT) Evaluation Criteria
  - Regent’s Traceability Log

**Release**

- **Entry Deliverables**
  - Unit Code
  - (UT) Authorization
  - (SIT) Authorization
  - (UAT) Plan
  - (SIT) Evaluation Criteria
- **Key Deliverables**
  - Unit Code
  - (UT) Evaluation Criteria
  - (SIT) Authorization
  - Regent’s Traceability Log
- **Exit Deliverables**
  - Unit Code
  - (UT) Evaluation Criteria
  - (SIT) Authorization
  - Regent’s Traceability Log
## IT Methodology

**SDLC – Systems Analysis Phase**

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Issue created PM deliverables</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req’s</td>
<td>Project Plan</td>
<td>Project Status Report</td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Schedule/ WBS</td>
<td>Team Status Report</td>
<td></td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Quality Assurance Plan</td>
<td>System Integration Test (SIT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Config, Management Plan</td>
<td>Change Request Definition</td>
<td>User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Project Procedures</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Systems Analysis

- **Entry Deliverables**
  - Business Case
  - Detailed Business Requirements

- **Exit Deliverables**
  - High Level Solution Design
  - Reg’mt’s Traceability Log

### Design

- **Entry Deliverables**
  - High Level Solution Design
  - Reg’mt’s Traceability Log

- **Exit Deliverables**
  - Detailed Solution Design
  - Programming Specifications
  - Reg’mt’s Traceability Log

### Build

- **Entry Deliverables**
  - Detailed Solution Design
  - Programming Specifications
  - Reg’mt’s Traceability Log

- **Exit Deliverables**
  - Testing Strategy
  - Reg’mt’s Traceability Log

### Test

- **Entry Deliverables**
  - Unit Test (UT) Plan
  - Evaluation Criteria
  - Reg’mt’s Traceability Log

- **Exit Deliverables**
  - (SIT) Plan
  - Evaluation Criteria
  - Reg’mt’s Traceability Log

### Release

- **Entry Deliverables**
  - Unit Code
  - (UT) Plan
  - Evaluation Criteria
  - Reg’mt’s Traceability Log

- **Exit Deliverables**
  - (UAT) Plan
  - Evaluation Criteria
  - Reg’mt’s Traceability Log
SDLC – Systems Analysis Phase

**Objectives**
- ensure a formal project exists and has been authorized *(Business Case established)*
- project team members review and understand project context *(deliverables)*
- identify/refine initial business requirements
- establish *(High Level Solution Design)*
- establish *(Requirements Traceability Log)*

**Iterations**
- iterative within the phase *(waterfall)*
- iterative within the lifecycle *(agile)*
Objectives

- ensure the application is designed in accordance with the authorized requirements defined during the System Analysis phase – (High Level Solution Design)
- complete the design of the application – (Detailed Solution Design)
- create specifications that reflect the authorized design & business requirements - (Programming Specification(s))
- maintain (Requirements Traceability Log)

Iterations

- iterative within the phase (waterfall)
- iterative within the lifecycle (agile)
# IT Methodology

## SDLC – Build Phase

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Create PM deliverables</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req’s</td>
<td>Project Plan</td>
<td>Project Status Report</td>
<td></td>
</tr>
<tr>
<td>Detailed Business Requirements</td>
<td>Project Schedule/WBS</td>
<td>Team Status Report</td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Deliverables</td>
<td>Unit Test (UT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Config, Management Plan</td>
<td>System Integration Test (SIT)</td>
<td></td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td>Project Procedures</td>
<td>Authorization</td>
<td></td>
</tr>
<tr>
<td>Project Procedures</td>
<td></td>
<td>User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
</tbody>
</table>

### Systems Analysis
- **Entry Deliverables**
  - Business Case
  - Detailed Business Requirements

### Design
- **Entry Deliverables**
  - High Level Solution Design
  - Reqmts Traceability Log

### Build
- **Entry Deliverables**
  - Detailed Solution Design
  - Programming Specifications
  - Reqmts Traceability Log

### Test
- **Entry Deliverables**
  - Unit Code
  - Test Plan
  - Reqmts Traceability Log

### Release
- **Entry Deliverables**
  - Unit Code
  - Reqmts Traceability Log
  - (UAT) Authorization
Objectives

- develop the required functionality according to the authorized design and business requirements
  (High Level Solution Design, Detailed Solution Design, Programming Specifications)
- create (Unit Code) of the application according to the programming specifications
- develop the (User Training and Support Plan)
- maintain (Requirements Traceability Log)

Iterations

- iterative within the phase (waterfall)
- iterative within the lifecycle (agile)
# IT Methodology

## SDLC – Test Phase

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Project Deliverables</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Project Plan</td>
<td>Project Schedule/WS</td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Deliverables</td>
<td>Project Plan</td>
<td></td>
</tr>
<tr>
<td>Config, Management Plan</td>
<td>Quality Assurance Plan</td>
<td>Team Status Report</td>
<td></td>
</tr>
<tr>
<td>Project Procedures</td>
<td></td>
<td>Project Status Report</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unit Test (UT) Authorization</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>System Integration Test (SIT) Authorization</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Systems Analysis
- **Entry Deliverables**
  - Business Case
  - Project Charter
  - Detailed Business Requirements

- **Exit Deliverables**
  - High Level Solution Design
  - Regent's Traceability Log

### Design
- **Entry Deliverables**
  - High Level Solution Design
  - Programming Specifications
  - Regent's Traceability Log

- **Exit Deliverables**
  - Detailed Solution Design
  - Regent's Traceability Log

### Build
- **Entry Deliverables**
  - Detailed Solution Design
  - Programming Specifications
  - Regent’s Traceability Log

- **Exit Deliverables**
  - Training & Support Plan
  - Regent’s Traceability Log

### Test
- **Entry Deliverables**
  - Unit Code
  - (UT) Evaluation Criteria
  - Regent’s Traceability Log

- **Exit Deliverables**
  - Unit Code
  - (UT) Evaluation Criteria
  - Regent’s Traceability Log

### Release
- **Entry Deliverables**
  - Unit Code
  - (UT) Authorization
  - Regent’s Traceability Log

- **Exit Deliverables**
  - Unit Code
  - (UT) Authorization
  - Regent’s Traceability Log
SDLC – Test Phase

**Objectives**

- perform all unit testing according to *(Unit Test Plans - from STLC)*
- Identify, log *(Unit Test Defect Log)* and address all conditions that do not satisfy unit testing criteria
- ensure the application/product has been built in accordance with the authorized design, business requirements, *(Unit Test Evaluation Criteria - from STLC)*
- ensure all unit testing deliverables/work products are under configuration management
- maintain *Requirements Traceability Log*

**Iterations**

- iterative within the phase (waterfall)
- iterative within the lifecycle (agile)
### IT Methodology

**SDLC – Release Phase**

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IT Methodology</strong></td>
<td><strong>SDLC</strong></td>
<td></td>
<td><strong>Release Phase</strong></td>
</tr>
</tbody>
</table>

**QAllassit Integrated Methodology**

**Programmed Deliverables**
- Business Case
- Detailed Business Requirements
- Project Charter
- Project Plan
- Project Schedule/WBS
- Roles & Responsibilities
- Project Deliverables
- Config. Management Plan
- Quality Assurance Plan
- Project Procedures

**Entry Deliverables**
- Business Case
- Detailed Business Requirements
- Project Charter
- Project Plan
- Project Schedule/WBS
- Roles & Responsibilities
- Project Deliverables
- Config. Management Plan
- Quality Assurance Plan
- Project Procedures

**Entry Deliverables**
- High Level Solution Design
- Regn’s Traceability Log

**Entry Deliverables**
- Detailed Solution Design
- Programming Specifications
- Regn’s Traceability Log

**Entry Deliverables**
- Unit Code
- (UT) Evaluation Criteria
- Regn’s Traceability Log

**Entry Deliverables**
- Unit Code
- (UT) Authorization
- Regn’s Traceability Log

**Entry Deliverables**
- Testing Strategy
- (UAT) Evaluation Criteria

**Entry Deliverables**
- (SIT) Plan
- (SIT) Evaluation Criteria

**Entry Deliverables**
- (UT) Plan
- (UT) Authorization
- Regn’s Traceability Log

**Entry Deliverables**
- (UT) Plan
- (UT) Authorization
- Regn’s Traceability Log

**Entry Deliverables**
- (UT) Plan
- (UT) Authorization
- Regn’s Traceability Log

**System Integration Test (SIT)**
- Authorization
- User Acceptance Test (UAT)
- Authorization
Objectives

- authorize that all necessary “unit” testing is complete - promote the newly built and unit tested functionality/application for additional testing (*Unit Test Authorization*)
- ensure all unit testing deliverables and work products under configuration management
- maintain *Requirements Traceability Log*
- migrate the application for additional (SIT, UAT) testing

Iterations

- iterative within the phase (waterfall)
- iterative within the lifecycle (agile)
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- **SDLC Phases - Objectives, Iterations, Traceability**
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- **SDLC Phases** - Objectives, Iterations, **Traceability**
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
## IT Methodology

### SDLC

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
</table>

### QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Systems Analysis</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
</table>
**IT Methodology**

**SDLC – Requirements Traceability**

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Project Plan</td>
<td>Team Status Report</td>
</tr>
<tr>
<td>Detailed Business Req’s</td>
<td>Project Plan</td>
<td>Project Plan Template</td>
<td>Status Report</td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Scope Statement</td>
<td>Project Schedule/Work Breakdown Structure</td>
<td>System Integration Test (SIT) Authorization</td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Quality Assurance Plan</td>
<td>Project Change Request Definition</td>
<td>User Acceptance Test (UAT) Authorization</td>
</tr>
<tr>
<td>Config. Management Plan</td>
<td>Project Task Log</td>
<td>Project Change Request Definition</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Project Procedures</td>
<td>Project Status Report</td>
<td>Project Change Request Definition</td>
<td>Project Closeout</td>
</tr>
</tbody>
</table>

**Entry Deliverables**

<table>
<thead>
<tr>
<th>Systems Analysis</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>High Level Solution Design</td>
<td>Detailed Solution Design</td>
<td>Unit Code</td>
<td>Unit Code</td>
</tr>
<tr>
<td>Detailed Business Requirements</td>
<td>Regret’s Traceability Log</td>
<td>Programming Specifications</td>
<td>(UAT) Plan</td>
<td>(UAT) Plan</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>Detailed Solution Design</td>
<td>Training &amp; Support Plan</td>
<td>Regret’s Traceability Log</td>
<td>Regret’s Traceability Log</td>
</tr>
<tr>
<td>In Level Solution Design</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Regret’s Traceability Log</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
</tbody>
</table>

**QAassist Integrated Methodology**
### IT Methodology

**SDLC – Requirements Traceability**

#### QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
</table>

#### Systems Analysis

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case&lt;br&gt; Detailed Business Requirements</td>
<td>High Level Solution Design&lt;br&gt; Req’ts Traceability Log</td>
<td>Detailed Solution Design&lt;br&gt; Programing Specifications&lt;br&gt; Req’ts Traceability Log</td>
<td>Unit Code&lt;br&gt; (UT) Evaluation Criteria&lt;br&gt; Req’ts Traceability Log</td>
<td>Unit Code&lt;br&gt; (UT) Evaluation Criteria&lt;br&gt; Req’ts Traceability Log</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td></td>
</tr>
<tr>
<td>High Level Solution Design&lt;br&gt; Req’ts Traceability Log</td>
<td>Programming Specifications&lt;br&gt; Req’ts Traceability Log</td>
<td>Unit Code&lt;br&gt; (UT) Evaluation Criteria&lt;br&gt; Req’ts Traceability Log</td>
<td>Unit Code&lt;br&gt; (UT) Evaluation Criteria&lt;br&gt; Req’ts Traceability Log</td>
<td></td>
</tr>
</tbody>
</table>

#### Software Development

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td></td>
</tr>
</tbody>
</table>

#### Software Testing

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td></td>
</tr>
</tbody>
</table>
### IT Methodology

**SDLC – Requirements Traceability**

#### QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initiate</strong></td>
<td><strong>Plan</strong></td>
<td><strong>Execute &amp; Control</strong></td>
<td><strong>Closeout</strong></td>
</tr>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Project Status Report</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Project Plan</td>
<td>Project Status Report</td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Schedule/WBS</td>
<td>Unit Test (UT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Project Risk Definition</td>
<td>System Integration Test (SIT) Authorization</td>
<td></td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td>Project Change Request Definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Procedures</td>
<td>Project Change Request Log</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Systems Analysis

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>High Level Solution Design</td>
<td>Detailed Solution Design</td>
<td>Unit Code</td>
<td>Entry Deliverables</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>High Level Solution Design</td>
<td>Detailed Solution Design</td>
<td>(UT) Evaluation Criteria</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Reg'n's Traceability Log</td>
<td>Programming Specifications</td>
<td>Reg'n's Traceability Log</td>
<td>(UT) Defect Log</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>Entry Deliverables</td>
<td>Testing Strategy</td>
<td>Testing Strategy</td>
<td>(UT) Authorization</td>
<td>Entry Deliverables</td>
</tr>
</tbody>
</table>
# IT Methodology

**SDLC – Requirements Traceability**

## QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Business Case</em></td>
<td><em>Project Charter</em></td>
<td><em>Project Closeout</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Project Plan</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Project Schedule/WS/S</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Roles &amp; Responsibilities</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Project Deliverables</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Config. Management Plan</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Quality Assurance Plan</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Project Procedures</em></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Initiate**
- Business Case
- Detailed Business Req's

**Plan**
- Project Charter
- Project Plan
- Project Schedule/WS/S
- Roles & Responsibilities
- Project Deliverables
- Config. Management Plan
- Quality Assurance Plan
- Project Procedures

**Execute & Control**
- Initially created PM deliverables
- Project Risk Definition
- Project Risk Log
- Project Issue Log
- Project Change Request Definition
- Project Change Request Log

**Closeout**
- Project Closeout

## Systems Analysis

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Design Deliverables</th>
<th>Build Deliverables</th>
<th>Test Deliverables</th>
<th>Release Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Business Case</em></td>
<td><em>High Level Solution Design</em></td>
<td><em>Detailed Solution Design</em></td>
<td><em>Unit Code</em> (UT) Plan</td>
<td><em>Unit Code</em> (UT) Plan</td>
</tr>
<tr>
<td><em>Detailed Business Requirements</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td>(UT) Evaluation Criteria</td>
<td>(UT) Evaluation Criteria</td>
</tr>
<tr>
<td><em>High Level Solution Design</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
</tr>
<tr>
<td><em>Detailed Solution Design</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
</tr>
</tbody>
</table>

## Test

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Test Deliverables</th>
<th>Release Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Unit Code</em> (UT) Plan</td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
</tr>
<tr>
<td><em>(UT) Evaluation Criteria</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
</tr>
</tbody>
</table>

## Release

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Test Deliverables</th>
<th>Release Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Unit Code</em> (UT) Plan</td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
</tr>
<tr>
<td><em>(UT) Evaluation Criteria</em></td>
<td><em>Regt’s Traceability Log</em></td>
<td><em>Regt’s Traceability Log</em></td>
</tr>
</tbody>
</table>

---

*QAassist 2.1*
Software Development Lifecycle (SDLC) Requirements Traceability

- User Requirements are dynamic – they are defined, clarified and authorized as project evolves
- User Requirements are calibrated (large to small) – calibration reflected in deliverables
- User Requirements are maintained throughout life of project – documented accordingly
- User Requirements created/established (by SDLC Phase)
  - Systems Analysis (Detailed Requirements, High Level Solution Design)
  - Design (Detailed Solution Design)
  - Build (Programming Specifications)
- User Requirements may be revised/ altered/ appended to during
  - Unit Testing
  - Integration Testing
  - User Acceptance Testing
Requirements Traceability Log (Matrix)  
Reference/Access/Updating

**SDLC**

- Systems Analysis Phase (Detailed Requirements, High Level Solution Design)
- Design Phase (Detailed Solution Design)
- Build Phase (Programming Specification)
- Test Phase (Unit Test Defect Log) ***
Establish webinar context - review previous “Concepts” sessions

- Jan - A Context for IT Methodology
- Mar - Apply PM Fundamentals to IT

Software Development Lifecycle (SDLC) – Context and Premise

SDLC - Requirements (Identification and Definition)

**SDLC Phases** - Objectives, Iterations, **Traceability**

SDLC – Internal Dynamics (PMLC and STLC interfaces)

Recap
Establish webinar context - review previous “Concepts” sessions

- Jan - A Context for IT Methodology
- Mar - Apply PM Fundamentals to IT

Software Development Lifecycle (SDLC) – Context and Premise

SDLC - Requirements (Identification and Definition)

SDLC Phases - Objectives, Iterations, Traceability

SDLC – Internal Dynamics (PMLC and STLC interfaces)

Recap
Internal Dynamics - SDLC Interfaces

- Project Management Lifecycle
  - Initial Project Planning (Project Plan and Project Schedule)
  - Ongoing Project Execution (Issue Log, Risk Log, Change Request, Status)

- Software Testing Lifecycle
  - Unit Test Planning, Execution & Authorization
Internal Dynamics - SDLC Interfaces

- **Project Management Lifecycle**
  - Initial Project Planning (Project Plan and Project Schedule)
  - Ongoing Project Execution (Issue Log, Risk Log, Change Request, Status)

- **Software Testing Lifecycle**
  - Unit Test Planning, Execution & Authorization
### Project Management with SDLC

**Initial Project Planning**

#### QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Project Risk Definition</td>
<td>Team Status Report</td>
</tr>
<tr>
<td>Detailed Business Req’s</td>
<td>Project Plan</td>
<td>Project Issue Log</td>
<td>Unit Test (UT) Authorization</td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Deliverables</td>
<td>Project Change Request Definition</td>
<td>System Integration Test (SIT) Authorization</td>
</tr>
<tr>
<td>Project Procedures</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Systems Analysis

<table>
<thead>
<tr>
<th>Software Development</th>
<th>Software Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Exit Deliverables</strong></td>
</tr>
<tr>
<td>Business Case</td>
<td>Unit Code (UT) Plan</td>
</tr>
<tr>
<td>Project Charter</td>
<td>(UT) Evaluation Criteria</td>
</tr>
<tr>
<td>Detailed Business Req’s</td>
<td>(SIT) Plan</td>
</tr>
<tr>
<td>Testing Strategy (UAT) Plan</td>
<td>(UT) Authorization</td>
</tr>
<tr>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Exit Deliverables</strong></td>
</tr>
<tr>
<td>High Level Solution Design</td>
<td>Unit Code (UT) Plan</td>
</tr>
<tr>
<td>Reqs’n Traceability Log</td>
<td>(UT) Evaluation Criteria</td>
</tr>
<tr>
<td>Detailed Solution Design</td>
<td>(SIT) Plan</td>
</tr>
<tr>
<td>Programming Specifications</td>
<td>(UT) Authorization</td>
</tr>
<tr>
<td>Reqs’n Traceability Log</td>
<td></td>
</tr>
<tr>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Exit Deliverables</strong></td>
</tr>
<tr>
<td>High Level Solution Design Testing Strategy</td>
<td>Testing Strategy (UAT) Plan</td>
</tr>
<tr>
<td>(SIT) Plan</td>
<td>(UT) Authorization</td>
</tr>
<tr>
<td>(SIT) Evaluation Criteria</td>
<td></td>
</tr>
<tr>
<td>Reqs’n Traceability Log</td>
<td></td>
</tr>
<tr>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Exit Deliverables</strong></td>
</tr>
<tr>
<td>Detailed Solution Design Programming Specifications</td>
<td>Detailed Solution Design Programming Specifications</td>
</tr>
<tr>
<td>Reqs’n Traceability Log</td>
<td>Reqs’n Traceability Log</td>
</tr>
<tr>
<td>Unit Code</td>
<td>Unit Code</td>
</tr>
<tr>
<td>Training &amp; Support Plan</td>
<td>Reqs’n Traceability Log</td>
</tr>
<tr>
<td>Reqs’n Traceability Log</td>
<td>Reqs’n Traceability Log</td>
</tr>
<tr>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Exit Deliverables</strong></td>
</tr>
<tr>
<td>Unit Code (UT) Plan</td>
<td>Unit Code (UT) Plan</td>
</tr>
<tr>
<td>(UT) Evaluation Criteria</td>
<td>(SIT) Plan</td>
</tr>
<tr>
<td>(SIT) Evaluation Criteria</td>
<td>(UT) Authorization</td>
</tr>
<tr>
<td>(SIT) Defect Log</td>
<td></td>
</tr>
<tr>
<td>Reqs’n Traceability Log</td>
<td></td>
</tr>
<tr>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Exit Deliverables</strong></td>
</tr>
<tr>
<td>Unit Code (UT) Authorization</td>
<td>Unit Code (UT) Plan</td>
</tr>
<tr>
<td>(UT) Evaluation Criteria</td>
<td>(UT) Evaluation Criteria</td>
</tr>
<tr>
<td>(SIT) Plan</td>
<td>(UT) Authorization</td>
</tr>
<tr>
<td>(SIT) Defect Log</td>
<td></td>
</tr>
<tr>
<td>Reqs’n Traceability Log</td>
<td></td>
</tr>
<tr>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Exit Deliverables</strong></td>
</tr>
<tr>
<td>Unit Code (UT) Authorization (SIT) Plan</td>
<td>Unit Code (UT) Plan</td>
</tr>
<tr>
<td>(UT) Evaluation Criteria</td>
<td>(UT) Evaluation Criteria</td>
</tr>
<tr>
<td>(SIT) Plan</td>
<td>(UT) Authorization</td>
</tr>
<tr>
<td>(SIT) Defect Log</td>
<td></td>
</tr>
<tr>
<td>Reqs’n Traceability Log</td>
<td></td>
</tr>
</tbody>
</table>

#### QAassist 2.1
Project Management with SDLC

Initial Project Planning

QAassist Integrated Methodology

Systems Analysis
- Entry Deliverables: Business Case, Detailed Business Need's
- Exit Deliverables: High Level Solution Design, Regn't's Traceability Log

Design
- Entry Deliverables: High Level Solution Design
- Exit Deliverables: Detailed Solution Design, Programming Specifications, Regn't's Traceability Log

Build
- Entry Deliverables: Detailed Solution Design
- Exit Deliverables: Unit Code, Training & Support Plan, Regn't's Traceability Log

Test
- Entry Deliverables: Unit Code
- Exit Deliverables: Unit Code, (UT) Evaluation Criteria, Regn't's Traceability Log

Release
- Entry Deliverables: Unit Code
- Exit Deliverables: Unit Code, (UT) Authorization, Regn't's Traceability Log

Project Management

Software Development

Software Testing
Internal Dynamics - SDLC Interfaces

- **Project Management Lifecycle**
  - Initial Project Planning *(Project Plan and Project Schedule)*
  - Ongoing Project Execution *(Issue Log, Risk Log, Change Request, Status)*

- **Software Testing Lifecycle**
  - Unit Test Planning, Execution & Authorization
Internal Dynamics - SDLC Interfaces

- **Project Management Lifecycle**
  - Initial Project Planning (Project Plan and Project Schedule)
  - **Ongoing Project Execution (Issue Log, Risk Log, Change Request, Status)**

- **Software Testing Lifecycle**
  - Unit Test Planning, Execution & Authorization
# Project Management with SDLC

## Ongoing Project Execution

### QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>All previously created PM deliverables</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Requirements</td>
<td>Project Plan</td>
<td>Project Risk Definition</td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Schedule/WS</td>
<td>Project Issue Log</td>
<td></td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Config. Management Plan</td>
<td>Project Change Request Definition</td>
<td></td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td>Project Procedures</td>
<td>Project Change Request Log</td>
<td></td>
</tr>
</tbody>
</table>

### Systems Analysis

**Software Development**
- **Entry Deliverables**
  - Business Case
  - Detailed Business Requirements

**Exit Deliverables**
- High Level Solution Design
- Regt's Traceability Log

### Design

**Entry Deliverables**
- High Level Solution Design
- Regt's Traceability Log

**Exit Deliverables**
- Detailed Solution Design
- Programming Specifications
- Regt's Traceability Log

### Build

**Entry Deliverables**
- Detailed Solution Design
- Programming Specifications
- Regt's Traceability Log

**Exit Deliverables**
- Unit Code
- Training & Support Plan
- Regt's Traceability Log

### Test

**Entry Deliverables**
- Unit Code
- (UT) Evaluation Criteria
- Regt's Traceability Log

**Exit Deliverables**
- Unit Code
- (UT) Evaluation Criteria
- Regt's Traceability Log

### Release

**Entry Deliverables**
- Unit Code
- (UT) Authorization
- Regt's Traceability Log

**Exit Deliverables**
- Unit Code
- (UT) Authorization
- Regt's Traceability Log
## Project Management with SDLC

### Ongoing Project Execution

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>All previously created PM deliverables</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Project Plan</td>
<td>Project Risk Log</td>
<td></td>
</tr>
<tr>
<td>Project Schedule/WBS</td>
<td>Project Schedule/WBS</td>
<td>Project Issue Log</td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Deliverables</td>
<td>Project Change Request Definition</td>
<td></td>
</tr>
<tr>
<td>Project Deliberables</td>
<td>Config, Management Plan</td>
<td>Project Change Request Log</td>
<td></td>
</tr>
<tr>
<td>Config, Management Plan</td>
<td>Quality Assurance Plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Procedures</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Systems Analysis

#### Entry Deliverables

- Business Case
- Project Charter
- Detailed Business Requirements

#### Exit Deliverables

- High Level Solution Design
- Reqmts. Traceability Log

### Design

#### Key Deliverables

- High Level Solution Design
- Programming Specifications

#### Entry Deliverables

- Detailed Solution Design
- Reqmts. Traceability Log

#### Exit Deliverables

- Detailed Solution Design
- Programming Specifications

#### Test

#### Entry Deliverables

- Unit Code (UT) Plan
- (UT) Evaluation Criteria

#### Exit Deliverables

- Unit Code (UT) Plan
- (UT) Evaluation Criteria

### Build

#### Entry Deliverables

- Detailed Solution Design
- Reqmts. Traceability Log

#### Exit Deliverables

- Detailed Solution Design
- Programming Specifications

### Test

#### Entry Deliverables

- Unit Code (UT) Plan
- (UT) Evaluation Criteria

#### Exit Deliverables

- Unit Code (UT) Plan
- (UT) Evaluation Criteria

### Release

#### Entry Deliverables

- Unit Code (UT) Plan
- (UT) Evaluation Criteria

#### Exit Deliverables

- Unit Code (UT) Plan
- (UT) Evaluation Criteria

---

(© QAassist 2.1)
Project Management with SDLC
Ongoing Project Execution
Project Management with SDLC
Ongoing Project Execution

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Case</strong>&lt;br&gt;Detailed Business Req's</td>
<td><strong>Project Charter</strong>&lt;br&gt;Project Plan</td>
<td><strong>Team Status Report</strong>&lt;br&gt;Project Status Report</td>
<td><strong>Project Closeout</strong>&lt;br&gt;Project Closeout and Final Report</td>
</tr>
<tr>
<td><strong>Project Schedule/WBS</strong>&lt;br&gt;Roles &amp; Responsibilities</td>
<td><strong>Project Plan</strong>&lt;br&gt;Project Schedule</td>
<td><strong>Unit Test (UT) Authorization</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Project Deliverables</strong>&lt;br&gt;Config. Management Plan</td>
<td><strong>Quality Assurance Plan</strong>&lt;br&gt;Project Procedures</td>
<td><strong>System Test</strong>&lt;br&gt;User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
<tr>
<td><strong>Project Risk Definition</strong></td>
<td><strong>Project Issue Log</strong></td>
<td><strong>System Test</strong>&lt;br&gt;User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
<tr>
<td><strong>Project Change Request Definition</strong></td>
<td><strong>Project Change Request Log</strong></td>
<td><strong>System Test</strong>&lt;br&gt;User Acceptance Test (UAT) Authorization</td>
<td></td>
</tr>
</tbody>
</table>

### Systems Analysis

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Case</strong>&lt;br&gt;Detailed Business Requirements</td>
<td><strong>High Level Solution Design</strong>&lt;br&gt;Reqmts' Traceability Log</td>
<td><strong>Detailed Solution Design</strong>&lt;br&gt;Programming Specifications</td>
<td><strong>Unit Code</strong>&lt;br&gt;(UT) Evaluation Criteria</td>
<td><strong>Unit Code</strong>&lt;br&gt;(UT) Authorization</td>
</tr>
<tr>
<td><strong>Exit Deliverables</strong>&lt;br&gt;High Level Solution Design</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Detail Solution Design</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Training &amp; Support Plan</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Unit Code</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Unit Code</td>
</tr>
<tr>
<td><strong>High Level Solution Design</strong>&lt;br&gt;Reqmts' Traceability Log</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Detail Solution Design</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Training &amp; Support Plan</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Unit Code</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Unit Code</td>
</tr>
</tbody>
</table>

### Software Development

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Case</strong>&lt;br&gt;Detailed Business Reqs</td>
<td><strong>High Level Solution Design</strong>&lt;br&gt;Testing Strategy</td>
<td><strong>Detailed Solution Design</strong>&lt;br&gt;Programming Specifications</td>
<td><strong>Unit Code</strong>&lt;br&gt;(UT) Authorization</td>
<td><strong>Unit Code</strong>&lt;br&gt;(UT) Authorization</td>
</tr>
<tr>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Testing Strategy</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;(SIT) Plan</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;(SIT) Evaluation Criteria</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;(SIT) Authorization</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;(SIT) Authorization</td>
</tr>
</tbody>
</table>

### Software Testing

<table>
<thead>
<tr>
<th>Entry Deliverables</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Business Case</strong>&lt;br&gt;Detailed Business Reqs</td>
<td><strong>High Level Solution Design</strong>&lt;br&gt;Testing Strategy</td>
<td><strong>Detailed Solution Design</strong>&lt;br&gt;Programming Specifications</td>
<td><strong>Unit Code</strong>&lt;br&gt;(UT) Authorization</td>
<td><strong>Unit Code</strong>&lt;br&gt;(UT) Authorization</td>
</tr>
<tr>
<td><strong>Exit Deliverables</strong>&lt;br&gt;Testing Strategy</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;(SIT) Plan</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;(SIT) Evaluation Criteria</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;(SIT) Authorization</td>
<td><strong>Exit Deliverables</strong>&lt;br&gt;(SIT) Authorization</td>
</tr>
</tbody>
</table>

**QAassist Integrated Methodology**

(QAassist 2.1)
Project Management with SDLC
Ongoing Project Execution
# Project Management with SDLC

## Ongoing Project Execution

## QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Project Risk Definition</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Project Plan</td>
<td>Project Status Report</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Project Schedule/WBS</td>
<td>Roles &amp; Responsibilities</td>
<td>Project Issue Log</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Config. Management Plan</td>
<td>Project Change Request Definition</td>
<td>Project Closeout</td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td>Procedures</td>
<td>Project Change Request Log</td>
<td>Project Closeout</td>
</tr>
</tbody>
</table>

## Systems Analysis

### Entry Deliverables
- Business Case
- Detailed Business Requirements
- High Level Solution Design
- Regnt's Traceability Log

### Exit Deliverables
- High Level Solution Design
- Regnt's Traceability Log

## Design

### Entry Deliverables
- High Level Solution Design
- Programming Specifications
- Regnt’s Traceability Log

### Exit Deliverables
- Detailed Solution Design
- Regnt's Traceability Log

## Build

### Entry Deliverables
- Detailed Solution Design
- Programming Specifications
- Regnt's Traceability Log

### Exit Deliverables
- Training & Support Plan
- Regnt's Traceability Log

## Test

### Entry Deliverables
- Unit Code
- (UT) Evaluation Criteria
- Regnt's Traceability Log

### Exit Deliverables
- Unit Code
- (UT) Authorization

## Release

### Entry Deliverables
- Unit Code
- (UT) Authorization
- Regnt’s Traceability Log

### Exit Deliverables
- Unit Code
- (UT) Authorization
- (UAT) Authorization

(© QAassist 2.1)
Internal Dynamics - SDLC Interfaces

- **Project Management Lifecycle**
  - Initial Project Planning (Project Plan and Project Schedule)
  - **Ongoing Project Execution (Issue Log, Risk Log, Change Request, Status)**

- **Software Testing Lifecycle**
  - Unit Test Planning, Execution & Authorization
Project Management Lifecycle

- Initial Project Planning (Project Plan and Project Schedule)
- Ongoing Project Execution (Issue Log, Risk Log, Change Request, Status)

Software Testing Lifecycle

- Unit Test Planning, Execution & Authorization
## SDLC with Software Testing Lifecycle (STLC)

Unit Test Planning, Execution & Authorization

### QAassist Integrated Methodology

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>All previously created PM deliverables</td>
<td>Team Status Report</td>
</tr>
<tr>
<td>Detailed Business Req’s</td>
<td>Project Plan</td>
<td>Project Risk Definition</td>
<td>Project Status Report</td>
</tr>
<tr>
<td></td>
<td>Project Schedule/WBS</td>
<td>Project Issue Log</td>
<td>Unit Test (UT) Authorization</td>
</tr>
<tr>
<td></td>
<td>Roles &amp; Responsibilities</td>
<td>Project Issue Definition</td>
<td>System Integration Test (SIT) Authorization</td>
</tr>
<tr>
<td></td>
<td>Project Deliverables</td>
<td>Project Change Request Definition</td>
<td>User Acceptance Test (UAT) Authorization</td>
</tr>
<tr>
<td></td>
<td>Config, Management Plan</td>
<td>Project Change Request Log</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Quality Assurance Plan</td>
<td>Project Procedures</td>
<td></td>
</tr>
</tbody>
</table>

### Project Management

- Business Case
- Detailed Business Requirements
- High Level Solution Design
- Regent’s Traceability Log

### Systems Analysis

#### Entry Deliverables
- Business Case
- Detailed Business Requirements

#### Exit Deliverables
- High Level Solution Design
- Regent’s Traceability Log

### Design

#### Entry Deliverables
- High Level Solution Design
- Regent’s Traceability Log

#### Exit Deliverables
- Detailed Solution Design
- Programming Specifications
- Regent’s Traceability Log

### Build

#### Entry Deliverables
- Detailed Solution Design
- Programming Specifications
- Regent’s Traceability Log

#### Exit Deliverables
- Unit Code
- Training & Support Plan
- Regent’s Traceability Log

### Test

#### Entry Deliverables
- Unit Code
- (UT) Evaluation Criteria
- Regent’s Traceability Log

#### Exit Deliverables
- Unit Code
- (UT) Evaluation Criteria
- Regent’s Traceability Log

### Release

#### Entry Deliverables
- Unit Code
- (UT) Authorization
- Regent’s Traceability Log

#### Exit Deliverables
- Unit Code
- (UT) Evaluation Criteria
- (UAT) Authorization

### Software Development

- Business Case
- Detailed Business Requirements

### Software Testing

- Business Case
- Detailed Business Requirements

- Testing Strategy
- (UAT) Evaluation Criteria
SDLC with Software Testing Lifecycle (STLC)
Unit Test Planning, Execution & Authorization
SDLC with Software Testing Lifecycle (STLC)
Unit Test Planning, Execution & Authorization
SDLC with Software Testing Lifecycle (STLC)
Unit Test Planning, Execution & Authorization
SDLC with Software Testing Lifecycle (STLC)
Unit Test Planning, Execution & Authorization
SDLC with Software Testing Lifecycle (STLC)
Unit Test Planning, Execution & Authorization
### SDLC with Software Testing Lifecycle (STLC)

#### Unit Test Planning, Execution & Authorization

<table>
<thead>
<tr>
<th>Initiate</th>
<th>Plan</th>
<th>Execute &amp; Control</th>
<th>Closeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Case</td>
<td>Project Charter</td>
<td>Project Closeout</td>
<td></td>
</tr>
<tr>
<td>Detailed Business Req's</td>
<td>Project Plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Schedule/WBS</td>
<td>Project Risk Definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roles &amp; Responsibilities</td>
<td>Project Issue Log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Deliverables</td>
<td>Project Change Request Definition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Config. Management Plan</td>
<td>Project Change Request Log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality Assurance Plan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Procedures</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**QAassist Integrated Methodology**

<table>
<thead>
<tr>
<th>Systems Analysis</th>
<th>Design</th>
<th>Build</th>
<th>Test</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Entry Deliverables</strong></td>
<td><strong>Entry Deliverables</strong></td>
</tr>
<tr>
<td>Detailed Business Requirements</td>
<td>Regmt’s Traceability Log</td>
<td>Programming Specifications</td>
<td>Regmt’s Traceability Log</td>
<td>(UT) Defect Log</td>
</tr>
<tr>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
<td>Exit Deliverables</td>
</tr>
<tr>
<td>High Level Solution Design</td>
<td>Detailed Solution Design</td>
<td>Detailed Solution Design</td>
<td>Unit Code</td>
<td>(UT) Defect Log</td>
</tr>
<tr>
<td>Regmt’s Traceability Log</td>
<td>Programming Specifications</td>
<td>Programming Specifications</td>
<td>Training &amp; Support Plan</td>
<td>Regmt’s Traceability Log</td>
</tr>
<tr>
<td></td>
<td>Regmt’s Traceability Log</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Exit Deliverables**

- SIT Plan (UAT) Evaluation Criteria
- SIT Plan (UT) Evaluation Criteria
- SIT Plan (UT) Authorization
- SIT Plan (UT) Defect Log
- Regmt’s Traceability Log

**QAassist 2.1**
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- SDLC Phases - Objectives, Iterations, Traceability
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
Leverage the Software Development Lifecycle Concepts III - Agenda

- Establish webinar context - review previous “Concepts” sessions
  - Jan - A Context for IT Methodology
  - Mar - Apply PM Fundamentals to IT
- Software Development Lifecycle (SDLC) – Context and Premise
- SDLC - Requirements (Identification and Definition)
- SDLC Phases - Objectives, Iterations, Traceability
- SDLC – Internal Dynamics (PMLC and STLC interfaces)
- Recap
Body of Knowledge

- Methodology/Lifecycle/Deliverables – www.qaiassist.com
  - Project Management, Software Development, Software Testing

- Body of Knowledge (Library) – www.qaiassist.com
  - Methodology Origin – origins of methodology – identifies the differences between the path we take (noun) and how we proceed (verb) down that path
  - Context & Overview – provides a general overview of IT Methodology and how it can be applied within an organization
  - Deliverable Descriptions – provides general descriptions for all the deliverables of the QAIassist Integrated Methodology
Keep Posing your Questions – call or send your questions in (Denise and/or Cameron)

Explore the QAlassist Body of Knowledge – refer to the QAlassist Body of Knowledge (sample deliverables, whitepapers, blog articles) – (www-qaiassist.com)

QAlassist Practitioner - find out more about becoming QAlassist certified “Foundation”, “Practitioner”

Tell your friends - tell your friends, coworkers and colleagues about the Project Insight/QAlassist webinars – invite them to join us at the next session
2017 Schedule

- The 3rd Wednesday of Every Month
  - 8:00 am Pacific - 11:00 am Eastern Time

- IT Methodology Concepts (60 mins)
  - This series of six webinars will be presented on a bi-monthly basis. Each webinar will provide you with a context, overview, terminology and general understanding of information technology (IT) methodology.

- IT Methodology – Tutorials (60 mins)
  - This series of five webinars will be presented on a bi-monthly basis. Each session is designed to offer a more “hands on” perspective – bridging the concepts into a practical utilization.
2017 “Concepts” Schedule

The 3rd Wednesday of Month (Jan, Mar, May, Jul, Sept, Nov)

- 8:00 am Pacific - 11:00 am Eastern Time

IT Methodology Concepts (60 mins)

- Methodology for the IT Landscape - (Jan)
- Apply PM Fundamentals to IT - (Mar)
- Leverage the Software Development Lifecycle - (May)
- **Explore the Software Testing Lifecycle** - (Jul)
- Incorporate Organizational Process Governance - (Sept)
- Implementing IT Methodology - (Nov)
2017 “Tutorials” Schedule

- The 3rd Wednesday of Month (Feb, Apr, Jun, Aug, Oct)
  - 8:00 am Pacific - 11:00 am Eastern Time

- IT Methodology Tutorials (60 mins)
  - Applying IT Methodology to Project Initiation - (Feb)
  - Project Planning & Design with IT in Mind - (Apr)
  - Executing Projects with IT Methodologies - (Jun)
  - Project Control & Verification - (Aug)
  - Project Close & Delivery - (Oct)
Moderator

Denise Rodriguez

Project Insight

Marketing

Denise.Rodriguez@projectinsight.com

www.projectinsight.net
Learn more!

🔹 Schedule a customized demo today
  ▪ +1 (949) 476-6499 x3
  ▪ info@projectinsight.net
  ▪ Request info: www.projectinsight.net

🔹 Contact QAassist
  ▪ +1 (613) 523-0052
  ▪ solutions@qaiassist.com
  ▪ Request info: www.qaiassist.com
Follow, like, subscribe!
Project Insight Community

- Sign up for more
- They are all FREE
- Check them out today!

- PM Training
- IT Methodology and Agile
- Leadership
- Product Training
See you next time!