Strategies for a Successful E2E Systems Integration Test

Fiona Charles
Let’s Test
May 9, 2012
This session

Describes key project management strategies I have used to manage large-scale Systems Integration Tests

Offers practical tips for implementing those strategies on your next SIT

The focus is primarily on large-scale integrations, but the strategies and tips can be scaled for any size of test.
1 – Project management
2 – Applications
3 – People
4 – Test readiness
5 – Information & communication
6 – Environments & data
7 – Questions & discussion
A Typical Systems Integration Project

Merchandising systems

New POS

Supply Chain systems

New Loyalty systems

Accounting systems

Point of Sale registers & Controllers

Management Information systems

Strategies for a Successful E2E SIT
The Test Manager’s Job

Develop a test strategy that addresses the major risks within the time available

Build the team you need for success

Build the right “controlled data” tests and plan

Manage testing and evaluation

Ensure acceptance by stakeholders

Sounds just like any other test - right???
The test management challenge is at least as great as the testing challenge

A project in itself

Always highly visible, and often controversial

Requirements for success often misunderstood

Your job is to hold to the big picture

*E2E Systems Integration Testing requires the skills of an experienced test project manager*
1 – Project management

2 – Applications

3 – People

4 – Test readiness

5 – Information & communication

6 – Environments & data

7 – Questions & discussion
It’s a Test Project Management job

Much of the complexity in managing a Systems Integration Test comes from the sheer number of things involved:

Applications
Platforms
Environments
Data
Application readiness dates
Business processes or functions
People

Managing these is a logistical, rather than test, challenge.
The numbers can seem overwhelming!
But it’s a nightmare only if you don’t manage it

Divide and conquer

Have a strategy for each type of thing, and a project plan that says when you are going to deal with it

Keep your priorities current

Don’t confuse the urgent with the merely important

Delegate where you can

Use project management tools and techniques
Be sure to level-set in your own head

Large Vehicles Need More Room

And then in everyone else’s head
Make sure you are positioned correctly

Senior role, working on equal footing with:
  - Project managers
  - Architects

No barriers to information
  - Pulse of the program
  - When each project is delivering (really)
  - Risks, issues, delays, individual test strategies, etc.

Authority
  - SIT entry criteria
  - Requirements to make SIT work

Reporting to the overall Project/Program Manager
1 – Project management
2 – Applications
3 – People
4 – Test readiness
5 – Information & communication
6 – Environments & data
7 – Questions & discussion
The applications will determine test scope

Start with the Architecture/integration diagrams
  Use whatever you can find
  If they don’t exist, draw your own
  Validate what you use

Keep an open mind about the test boundaries
  “Received belief” may be wrong
  Focus on “whole system” outcomes
  Identify all incoming data that could influence your results
  Ask what the end point should really be
    What’s the last application or report the critical data impacts?
    Who needs to sign off?
  You may need to include unchanged systems
1 – Project management
2 – Applications
3 – People
4 – Test readiness
5 – Information & communication
6 – Environments & data
7 – Questions & discussion
You will be dealing with LOTS of people
The rest of the project management team
  Project or Program Manager
  Architects
  Project Control

People associated with each application, e.g.,
  Project leads
  Project test leads and testers
  Programmers/Architects

Management
  People outside the project who will want information

QA, auditors, business sponsors, acceptors

Your own test team
# Make sure you have a strategy for each group

| The rest of the project management team | • Build relationships - lunch  
| | • Status reports, regular status meetings |
| People associated with each application | • Test working group or forum – get input & buy-in for strategy  
| | • SIT progress broadcasts |
| Management | • 1 on 1’s  
| | • Occasional status/strategy presentations |
| QA, auditors, business sponsors, acceptors | • Approach them – know who they are and what they need |
Success depends on building relationships

Be aware of cultural differences
  Different technology groups may have very different processes
  Different groups may have varied risk tolerances: e.g., front-line business vs. finance or audit
  Distributed teams in different countries

Understand what each group wants/needs from your tests

Work to get early buy-in for everything significant in your plan
  Develop your test strategy in a lightweight medium, and walk it around (“socialize” it)
  If you have to publish a detailed document, try to get consensus before you write it
Staff your own team appropriately
You won’t have time to be hands-on!

Strong second-in-command
Day-to-day SIT test team management

Environment and data manager
Work with all participating teams

Small team of seasoned testers
Design tests; prepare test materials

Toolsmith(s)
Any automation you can do
1 – Project management
2 – Applications
3 – People
4 – Test readiness
5 – Information & communication
6 – Environments & data
7 – Questions & discussion
Establish the ground rules (Entry & exit criteria)

**Environments** ready, connected and tested

Coordinated **data**

**Project-level tests** complete, e.g.,
  - System and usability test
  - Performance
  - Near-neighbor interfaces

All must-fix **bugs**, and all integration bugs of any severity, resolved

**Teams** assigned (and set up in defect tracking system)

**Communicate requirements and get early buy-in**
Track readiness

Your own team’s deliverables
- Test materials (test cases and scenarios)
- Environments
- Data
- Initial setup of defect tracking system

System readiness
- All participating systems
- Readiness to meet SIT entry criteria

Report on readiness weekly using an easy-to-understand graphic
A simple timeline for system readiness

**Assumptions**
1. Main App v0.9 will be full-featured and stable.
2. Project 3 Apps 2 & 3 will be E2E-ready no later than 12-Oct.
3. Starting E2E test without Project 3 Apps 4 & 5 will be feasible and will provide E2E test value.

**Risks**
1. Required modifications to GL Journalling and back-end financial accounting have not yet been finalized or planned. (Change Request #29)
Verify readiness

Schedule a handover meeting with each team before scheduled entry to SIT

Ask to see evidence (give advance notice of what that should be)

Give teams a standard checklist to populate

Review the checklist and evidence with each team when you meet
# Readiness Documentation Examples

<table>
<thead>
<tr>
<th>Test requirement</th>
<th>Criteria</th>
<th>Sample Documentation</th>
</tr>
</thead>
</table>
| System/functional test | Evidence of testing covering the functionality delivered in the release | • Test cases & traceability matrix  
• Business acceptor sign-off on test coverage  
• IDs & detail for outstanding bugs  
• Exploratory test documentation (charters, logs. Etc.)  
• IDs & detail for outstanding bugs |
| Regression test of systems currently in production and modified for this project | Completed regression test covering critical production functionality. | • Evidence that the required testing has been done (risk assessment, test completion checklist/report, etc.)  
• Sign-off by required IT and/or business acceptors. |
## Readiness Documentation Examples

<table>
<thead>
<tr>
<th>Test requirement</th>
<th>Criteria</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>UAT (or Agile Acceptance Test)</td>
<td>Completion of (User) Acceptance Test covering business acceptance criteria.</td>
<td>• Business acceptor sign-off OR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Conditional acceptance by business acceptors, pending final acceptance at the end of SIT.</td>
</tr>
<tr>
<td>All project test levels</td>
<td>Outstanding bugs:</td>
<td>• Summary report showing current status of all bugs logged during testing</td>
</tr>
<tr>
<td></td>
<td>• No outstanding must-fix</td>
<td>• Detailed bug reports for outstanding bugs, with analysis of downstream impacts, documented workarounds, and resolution plans</td>
</tr>
<tr>
<td></td>
<td>• Review outstanding bugs with impacts on downstream data or function</td>
<td></td>
</tr>
</tbody>
</table>
1 – Project management
2 – Applications
3 – People
4 – Test readiness
5 – Information & communication
6 – Environments & data
7 – Questions & discussion
Keep yourself informed

Take advantage of all the communication channels

Make sure you are in the key program/project level status meeting

Ask regularly for updates on system readiness

Review the program/project risk and issue logs for anything that might affect your test

Chat to project managers and test leads
Build your network
Outward communication

Make regular presentations to all the stakeholder groups
  
  SIT strategy
  
  Progress reports

Set up a “notice-board” to post daily status (you’ll need this during E2E SIT execution)
  
  Intranet, etc.

Set up a “test question and answer” forum for your team’s questions and share it on the notice-board
1 – Project management
2 – Applications
3 – People
4 – Test readiness
5 – Information & communication
6 – Environments & data
7 – Questions & discussion
Strategies for a Successful E2E SIT
Some of your most difficult issues are here

Multiple platforms (hardware, OS, DBMS)

Other infrastructure, like batch schedulers and message transports

Exclusive use for your test of adequate test environments

Data, co-ordinated across participating applications

Configuration management

Co-ordinated defect management

You need an Environment/Data Manager!
Environment/Data Manager

Inventory all applications and infrastructure elements

Work with you and SIT test team on data strategy

Work with each team to develop plans
   Environment plan
   Data plan

Develop configuration management process

Administer configuration management process during E2E test execution
Test Environments

Virtualize if you can – but you may not be able to

Physical environments may not exist for every application

There may not be connectivity where you need it

You may have to negotiate for exclusive use

If setup/purchase is required, there will be long lead times

Start early! (Get that Environment Manager on it!)
Coordinated Test Bed Data

This can be very difficult to achieve

A showstopper if you don’t have it

You need a strategy and a plan

Start early! (Get that Environment Manager on it!)
Configuration Management

Can be complex, with Production fixes and other projects going on at the same time as yours

You need an exact picture at all times

A system solution across platforms will probably not work

Design a simple “logical release” process:

Start with an inventory of all the systems, platforms, infrastructure elements participating in the test

Names, versions, etc.

Logical Release 1 is the starting inventory

Any time ANYTHING changes: application for defect resolution, infrastructure upgrade, etc., update the inventory, and you’re now on Release 2
Defect Management

Essential to have one defect management system for the entire integration

Set it up so you can track by application /project/ vendor

Set up a SWAT team to handle defect diagnosis and assignment
   Neutral lead, e.g., overall Project Architect
   Representative from each team

Make the Lead a regular attendee at your daily checkpoint during test execution
Use project management to go from this
To something like this
Questions & discussion