Managing Software Projects With Intelligence

Paul Gerrard
Gerrard Consulting
1 Old Forge Close
Maidenhead
Berkshire
SL6 2RD UK

e: paul@gerrardconsulting.com
w: http://gerrardconsulting.com
t: +44 (0)1628 639173

Paul Gerrard

Paul is the founder and Principal of Gerrard Consulting, a services company focused on increasing the success rate of IT-based projects for clients. He has conducted assignments in all aspects of Software Testing and Quality Assurance. Previously, he has worked as a developer, designer, project manager and consultant for small and large developments using all major technologies and is the webmaster of gerrardconsulting.com and several other websites.

Paul has degrees from the Universities of Oxford and London, is Web Secretary for the BCS SIG in Software Testing (SIGIST), Founding Chair of the ISEB Tester Qualification Board and the host/organiser of the UK Test Management Forum conferences. He is a regular speaker at seminars and conferences in the UK, continental Europe and the USA and was recently awarded the “Best Presentation of the Year” prize by the BCS SIGIST.

Paul has written many papers and articles, most of which are on the Gerrard website. With Neil Thompson, Paul wrote “Risk-Based E-Business Testing” – the standard text for risk-based testing.
Why Projects Fail
Why projects fail

- Won’t bore you with yet another survey of IT projects that fail - take it for granted, most do
- But why is this?
- Hundreds of modes of project failure – all very well documented
- But a large proportion of project failures can be traced to poor decision making:
  - Constraints to decision making – “you are not allowed to decide”
  - Decisions already made cannot be changed
  - Decisions that are ‘too big’ to make now – it’s simply too late
- Sometimes, the worst decisions are those that are never made at all.

Decision-making when things go wrong

- Imagine a world where the project plan is perfect, and everything goes well
- There are few decisions to make except exit/acceptance reviews
  - These would be easy to make, wouldn’t they?
- But plans aren’t perfect and things don’t go to plan
- Most decision-making is required when things don’t go to plan
  - Late delivery – we are late or expect to be late
  - Poor quality or stability – deliverables not meeting requirements, fail to stabilise or are defective
  - Technical challenges that cannot be met – technology doesn’t work, can’t demonstrate working systems, cannot achieve objectives with proposed solution.
Decision-making dependent on good information

- When called upon to make a call on delays, quality or solution viability...
  - Stakeholders and project managers usually lack the right information at the right time
  - Most of the information actually required is TEST EVIDENCE
- Early and continuous review and testing will highlight
  - Items not ready (we can't test them)
  - Poor quality (they are buggy)
  - Developers that cannot deliver
- I call this evidence Project Intelligence
- THE TEST STRATEGY IS A STRATEGY FOR GATHERING PROJECT INTELLIGENCE.

Our project management approach is critically flawed

- The way we plan projects and monitor/manage progress is flawed
- Project plans set out a prediction of the future
- A network of interdependent activities aligned with a calendar
- We identify activities, costs and timelines
- We monitor activities, spend and the calendar
- We act when progress doesn't meet the plan.
But STOP! Imagine we planned a journey this way

- We monitor progress by checking...
  - The time – “it’s 12.45pm – we must be half way”
  - The fuel gauge – “we are nearly out of gas – we must be there!”
- This is nonsense of course
- We are monitoring INPUTS, NOT OUTPUTS
- We should be monitoring ACHIEVEMENTS, not ACTIVITY
- What are achievements?
- How do we monitor them?

The measurement of achievement

- The achievements of most significance are
  - Deliverables (on time and of good quality)
  - Decisions made (on the acceptability of deliverables)
- But wait! The REAL measure of achievement is the acceptance decision, based on evidence
  - Requirements
  - Designs
  - Code
  - Systems
  - Etc…
- Testing provides that evidence.
Pawnbroker model of IT development

- Developers usually complain that requirements change
- Were the users focused on the business goal?
- Developers focus on technical deliverables, as usual
- But who keeps them on the right track?

- Required, designed, not delivered (cuts through deadline pressure?)
- Not required, designed, delivered (an analyst's bright idea?)
- Required, not designed, delivered (the user bribed a programmer?)

Four-eyed plans

Increasingly Inaccurate IT-focused Initial Plans

The plan is a model of the project. The real project consists of people, organisation, goals and risks.
Testing provides the “intelligence”

- Our deliverables are intermediate project goals
  - Are they available? Are they acceptable? Do they meet requirements?
- Testing can answer these questions
  - No tests completed? The deliverable isn’t available
  - Test successes/failures indicate quality and whether deliverable meets requirements
- We manage product risks and goals, not activities:
  - We analyse the risks associated with this deliverable
  - We plan tests to address uncertainty
  - We monitor tests planned, passed and failed.

Poor contracts result in poor plans and poor project management

- Inputs to project plan for delivery
  - The Contract
  - The Requirements (business or technical level)
- The requirements may be good or bad
- But if the contract is poor or inflexible
  - Things get forgotten, underestimated
  - Testing, risk management, reporting all suffer
  - Project Management attention on timescales and costs (inputs) not on deliverables, quality, benefits (outputs)
  - Acceptance process and criteria are loose, so plans are woolly when precision is most important.
Introduction to Project Intelligence

Testing throughout the life cycle: the W model
Almost all project activities generate deliverables
Project goals depend on deliverables
Every deliverable has an associated test activity
  - Static tests: reviews, inspections, static analysis etc.
  - Dynamic tests: component, link, system, acceptance tests etc.
Testing provides the evidence of achievement
  - Of intermediate goals throughout a project
  - Of final project goal
(This is standard PRINCE approach, but is appropriate for all methodologies, by the way).

Three types of risk

(Logistics, Planning and Control)
Management Domain
Few, easy to understand, manage

Deliverables
Practitioner Domain
Many, hard to understand, manage
What is a test?

Testing doesn’t change the product in any way. We do it to gather ‘intelligence’.

Testing generates “project intelligence”

- If you don’t know where you are, a map won’t help
- If you don’t know the status of the deliverables, the best plan won’t help you manage your project
- A project is like driving off-road in the dark
  - Testing provides the headlights for the journey
- Testing provides data on the status of deliverables and generates “PROJECT INTELLIGENCE”
- Stakeholders and management need intelligence to make decisions, command and control the project.
Project Intelligence starts with goals

- What are our project goals?
- Our goals depend on the deliverables
  - Which are the key deliverables?
  - What are the dependencies between them?
- How could our deliverables fail?
  - Evidence required to be sure they won’t fail?
- How can I be sure our goals are met?
  - Evidence required to demonstrate they are met?
Key sources of project intelligence

- The value of project intelligence depends on how well we achieve and communicate:
  1. Coverage – planned and achieved
  2. Product Risks – and the status of those risks
  3. Business Goals – and the status of those goal
  4. Test Status – passes, failures, incidents

- Time to deliver intelligence depends on the test process

- Willingness of the organisation to accept and act upon project intelligence is key.

Based on established disciplines

- The PI approach uses a combination of established techniques (you might know one or two already!)
  - Results-Based Management
  - Results Chain Analysis (modelling)
  - Benefits Realisation
  - Performance Measurements
**PI Techniques**

- Results Chain Analysis
- PI Strategy
- Test Strategy Completion
- (Assessment and Testing)
- PI Reporting

---

**PI and the project lifecycle**

<table>
<thead>
<tr>
<th>Key:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Project planning and initiation</td>
<td>Project Intelligence Activities</td>
</tr>
<tr>
<td>Development activities</td>
<td>Review and Test activities</td>
</tr>
</tbody>
</table>

---

© 2003-7 Gerrard Consulting Limited

Version 1.0
Results Chain Analysis

RCA defines the relationship between goals, activities and risks.
Goal, Benefits and deliverables

Technology | Features | Change in Work Practices | Benefits | Business Goals

Project Deliverables

Electronic communication with suppliers
Automatic re-ordering
System accessible to all requisitioners
Expedition
Closer working with suppliers
Electronic entry
Electronic information
Improved supplier understanding
More timely procurement
Improved co-ordination between functions
More accurate deliveries
More timely deliveries
Optimised selection of suppliers

To reduce production downtime

Administrative By-products
Spending the budget, meeting deadlines, filling timesheets, compiling RAG reports, ticking boxes, endless meetings

Waste Products?

Business Results
Achievement of business goals as a consequence of the investment in the programme/project.

The Results Chain
Results Chain Analysis (RCA)

- Takes place as part of the overall project planning process
- Intended to document the relationship between:
  - Activities (initiatives)
  - Deliverables (outcomes)
  - Dependencies (contributions)
  - Risks and assumptions (assumptions)
  - Intermediate and final project goals (outcomes again)
- Used to identify risks early and tie activities to goals and risks.

RCA Diagram elements

- Outcome - an output (deliverable), outcome (intermediate goal) or impact (final goal) of the project
- Initiative - an activity intended to create or acquire a deliverable, achieve a desired outcome etc.
- Contribution - identifies a contribution of an initiative, assumption or outcome to another initiative, or outcome.
- Assumption - identifies a dependency on an external situation, not usually under the control of the project
Using RCA

- RCA models the benefits realisation process
- Stakeholders and management can see the road map leading to the business goal
- Derived from interviews and workshops
- Promotes discussion, consensus and commitment
- A shared understanding of the linkages between business and IT initiatives
- Makes implicit thinking explicit, brings hidden assumptions to the surface and enables better communication and decision making.
PI Strategy generates the assessment process
Use goals and risks to define the process for intelligence gathering

- Test process supports the intelligence gathering process
- Cannot rely on fixed, inflexible test processes (but we keep the best of ‘best practice’)
- Test process design driven by the need to report progress against goals and risk
- NB: coverage goals are included as usual
- NB2: we will need new forms of assessment beyond traditional static/dynamic testing.

Goals and risks as PI Drivers

- Strategy for goal-based progress reporting:
  - Intermediate and final business goals
  - Deliverables: documents, software, infrastructure, training, resources etc.
  - Coverage goals: functional/non-functional requirements, scenarios, regulatory requirements etc.
  - “To demonstrate that these goals have been met or identify the problems that block the goal”

- Strategy for risk-based progress reporting
  - Product risks: “to demonstrate that risks have materialised, or have been addressed”.
Project Intelligence Strategy

<table>
<thead>
<tr>
<th>PI Drivers</th>
<th>Ass. Obj.</th>
<th>Project Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business goals</td>
<td></td>
<td>Reqs</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Objectives for each test phase are easily identified.
Test Strategy Completion

- E.g. System test...
  - Objectives (from PI strategy)
  - Environment (technology, mainly)
  - Responsibility (plan, spec, execution, approval)
  - Deliverables (plan, spec, incidents, reports)
  - Techniques (test design, intelligence reporting)
  - Tools (plan, spec, execution, data prep, reporting etc.)
  - Entry/exit criteria (goals met, risks addressed)
  - Etc. etc.
Test Strategy Completion 2

- Coverage policy
- Timescales/estimates (or the policy for estimation)
- Regression testing policy
- Incident management process
- Reporting approach
- Usual stuff…etc. etc.

PI Process – designed to handle change
Goal, Risk and Requirements Based Testing

Requirements Coverage
– Typical System Test

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Customers</td>
<td>X</td>
</tr>
<tr>
<td>Manage Products</td>
<td></td>
</tr>
<tr>
<td>Create Requisition</td>
<td></td>
</tr>
<tr>
<td>Authorise Requisition</td>
<td></td>
</tr>
<tr>
<td>Raise Sales Order</td>
<td></td>
</tr>
<tr>
<td>Allocate Stock</td>
<td></td>
</tr>
<tr>
<td>Fulfil Order and Distribute</td>
<td></td>
</tr>
<tr>
<td>Raise Invoice</td>
<td></td>
</tr>
<tr>
<td>Match Payment</td>
<td></td>
</tr>
<tr>
<td>And so on…</td>
<td></td>
</tr>
</tbody>
</table>

Large number of tests, each focusing on a limited scope
Requirements Coverage
– Typical User/End to End Test

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Customers</td>
<td>X</td>
</tr>
<tr>
<td>Manage Products</td>
<td>X</td>
</tr>
<tr>
<td>Create Requisition</td>
<td>X</td>
</tr>
<tr>
<td>Authorise Requisition</td>
<td>X</td>
</tr>
<tr>
<td>Raise Sales Order</td>
<td>X</td>
</tr>
<tr>
<td>Allocate Stock</td>
<td>X</td>
</tr>
<tr>
<td>Fulfil Order and Distribute</td>
<td>X</td>
</tr>
<tr>
<td>Raise Invoice</td>
<td>X</td>
</tr>
<tr>
<td>Match Payment</td>
<td>X</td>
</tr>
</tbody>
</table>

Smaller number of tests, each focusing on a business process path

Product risks as potential modes of failure

- Product risks are speculative failures and incidents
- The risk impact scale is the SAME as incident severity
- Some modes of failure are unacceptable so we select tests to demonstrate selected failure modes are unlikely
  - NB: these tests COULD be in any project phase
  - Means we focus on the tests that MUST pass
  - We know defects that cause these failures MUST be corrected
- If we can’t correct failures of risk-based tests we can say these risks have materialised
- Critical risks block acceptance
  - Force a change of plan: re-schedule, redesign, re/de-scoping or abandonment!
Risk-Based Testing complements requirements-based

<table>
<thead>
<tr>
<th>Product Risks</th>
<th>Test Obj.</th>
<th>Project Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reqs</td>
<td>Design</td>
</tr>
<tr>
<td>High</td>
<td>Build</td>
<td>Integ</td>
</tr>
<tr>
<td></td>
<td>Systest</td>
<td>UAT</td>
</tr>
<tr>
<td></td>
<td>Trial</td>
<td>Prod.</td>
</tr>
<tr>
<td>Low</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Objectives for each test phase are easily identified.

Risk-based reporting

Start today

Residual Risks

All risks 'open' at the start

Progress through the test plan
Goal-Based Testing completes the picture – Project Intelligence

- All projects have underpinning goals/benefits
- Stakeholders want to know:
  - The status of these goals
  - The status of the risks that block those goals
  - The confidence with which they may make a decision (test coverage)
- These are the three areas where test evidence is required: Project Intelligence.

Project Intelligence and Coverage

The tests in scope cover requirements, risks and goals.

Any test may be designed to address the needs of one, two or all three drivers.

Corollary: a test that doesn’t address any driver is a pointless test.
## Project Intelligence Strategy

<table>
<thead>
<tr>
<th>PI Drivers</th>
<th>Ass. Obj.</th>
<th>Project Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business goals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Project Phase

<table>
<thead>
<tr>
<th>Reqs</th>
<th>Design</th>
<th>Build</th>
<th>Integ</th>
<th>Systest</th>
<th>UAT</th>
<th>Trial</th>
<th>Prod.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Objectives for each test phase are easily identified.*

### PI reporting

<table>
<thead>
<tr>
<th>PI Drivers</th>
<th>Ass. Obj.</th>
<th>Project Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risks</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Project Phase

<table>
<thead>
<tr>
<th>Reqs</th>
<th>Design</th>
<th>Build</th>
<th>Integ</th>
<th>Systest</th>
<th>UAT</th>
<th>Trial</th>
<th>Prod.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Progress of achievement is clearly seen. Outstanding objectives are highly visible.*
Goal-based test reporting

Goal achieved, Benefits available

Close
PI Summary

- Historically, coverage (requirements, code etc.) was the only game in town
- Risk-based introduced a second dimension
- Goal-based is the third
- Testing is transformed from just defect-detection to intelligence gathering and decision-support
- Project Intelligence
  - Extends from business case through to post deployment assessment
  - Involves document, code, system, user, service testing
- Not necessarily more tests, but more focused tests, much better control and stakeholder reporting.

Finally

- Best PM practice: manage goals, not activities
- PI gives management visibility of risks, goals and progress towards them
- Project managers and stakeholders are PI customers
- The test process is the nervous system of your project, designed to provide useful PI in a timely manner
- What’s new?
  - Goal and risk-based testing
  - New styles of test reporting throughout the lifecycle
  - Appreciation of testing as a key source of Project Intelligence
- PI is usable in ALL methodologies.
Managing Software Projects With Intelligence

Thank-You!

gerrardconsulting.com
uktmf.com