Estimation Techniques for Testing Projects

V.K. Kripanand
Table of Contents

- Estimation
  - Overview
  - Components of Estimation
  - Factors for Estimation
  - Estimation Model
  - Techniques in Use
  - A discussion on techniques
  - Key Points and Future forward

- Test Projects Estimation
  - Why now?
  - How is it Different?
  - Why is it Important?
  - Future Forward
Table of Contents

- Estimation Techniques for Testing Projects
  - Principles of Estimation
  - Black Box
  - White Box
  - Performance Testing
  - Stress Testing
  - Other Testing
  - Influencing Factors
  - Team Dynamics

- Worksheets to use
  - Test Estimation Worksheets
  - Other Supplemental Worksheets

- Summary
Estimation

Topics

- Overview
- Components of Estimation
- Factors for Estimation
- Estimation Model
- Techniques in Use
- A discussion on techniques
- Key Points and Future forward
Estimation

Overview

- Current Estimation techniques are aplenty
- No subscribed method adopted
- Experience, plays a crucial role
- “Plug and Pray” – “Predict and Pray”
- Why do we need estimation at all?
Components of Estimation

- Manpower cost
- Cost of infrastructure
- Operational costs
  - Special software
  - Communication
  - Travel
  - Training
- Overheads
  - Shared facilities
  - Infrastructure overheads
  - Cost of money
  - Cost of Electricity/welfare/overtime etc
Factors for estimation

- Market opportunity
  - Who the competition is?
  - What is the opportunity now and in the future?
- Contractual terms
  - Are there penalty clauses?
  - Is it phase wise delivery?
- Volatility of requirements
  - How standard is the requirement?
  - What could be the change in requirements over time?
- Past Experience
  - Have we had a better or bitter experience?
  - Have we delivered something similar in the past?
Factors for estimation

- Organizational strengths
  - Management support
  - Strength and expertise of the team
- Skill set
  - Talent pool of engineers and the expectations of the project
  - In house training capability
  - Ability to learn quickly
- Ability to ramp up
  - Can we ramp up our team in case we win the project?
  - Option and flexibility to outsource?
- Technology
  - Complexity
  - Environment – Onsite/Offsite
  - Virtual Testing
Estimation Models

- What is it?
  - Estimation model is the combination of the technique and the factors
  - A predefined framework or template
  - Adaptable with modifications

- Build or use Models?
  - Using existing models is a safe bet
  - Derive and build your own models
Techniques in use

- Pricing to Win
  - What the customer wants to spend
  - Based on budget not on functionality
- Parkinson’s law
  - Expand to fill time
  - No objective assessment
- Analogy
  - Domain based
  - Hugely dependent on past completed projects.
- Expert Judgment
  - Experts of the domain and development practice are consulted
  - Agreement on final estimate is an iterative process
- Algorithmic cost modelling
  - Based on historical cost information
  - Software metric information and a model determine the effort
Some more...

- Pricing to win, Parkinson’s law, Expert judgment
  - May not be relevant in today’s context
- Analogy and Algorithmic models
  - Widely in use
- COCOMO and its variants are a great example
  - Algorithmic approach
  - Based on metrics and formulae
  - Repeatable and Re-usable
- Function points, object points
  - Provide a framework based input
Some more...

- Estimation is not only about effort
  - Model reveals the importance of staff experience in software development
  - As well as effort estimation, managers must estimate the calendar time
  - The time required is independent of the number of people working on the project
  - Staff required $<> (development time/required schedule)$
  - Phase of the project decides the people working on the project

- Cost/schedule risk management
  - Perform computations with and without the risk being realized and see the size of the impact $\rightarrow$ best/worst case estimates
A discussion on techniques...

- Each method has strengths and weaknesses
- Estimation should be based on several methods
- If these do not return approximately the same result, there is insufficient information available
- Some action should be taken to find out more in order to make more accurate estimates

Challenges
- When RS is incomplete
- Complete Details are not available
- Too frequent changes to requirements
Key Points and future forward

- Which Model to use?
  - Because no model is right, all models can be useful
  - Different techniques of cost estimation should be used
- Factors that influence
  - Productivity, Individual aptitude
  - Domain experience
  - Project size, Tool support and Technology challenges
- Use of specific models
  - Algorithmic cost estimation is difficult because of the need to estimate attributes of the finished product
  - The COCOMO model takes project, product, personnel and hardware attributes into account when predicting effort required
  - Algorithmic cost models support quantitative option analysis
  - The time to complete a project is not proportional to the number of people working on the project
Key Points and future forward

- Future Forward
  - Put the best models in use and derive a standard template
  - Learn from past experiences and document the same
  - Collect metrics to use in future, not for name sake
  - Junk in – Junk out (Actual effort data is very crucial)
  - With over 2 decades of collective experience, we should now retrospect with past data
  - Collaborate and co-operate to have a win-win
  - Cut across competitive lines…
  - Adapt the same for different nature of projects – More about this later……
Test Projects Estimation

- Topics
  - The three W’s!
  - Why now?
  - How is it Different?
  - What is Important?
  - Future Forward
Test Projects Estimation

- The three W’s

  - Why?
    - Why the focus now on Test Projects?
  
  - How?
    - How is this different from normal projects?
  
  - What?
    - What are the important parameters to consider?
Why Now?

- The market
  - Gartner estimates the worldwide software testing market at $13 Billion
  - India is expected to touch $1 Billion by 2007

- What we need now?
  - Established Estimation Techniques
  - Build large and skilled teams
  - Remove the fad of ‘Testing’ as a non-interesting job!
  - Build frameworks and process methods specific to testing
  - Use automation effectively
  - Leverage the large talent pool
  - Incubate talent with the relevant skills
How is it Different?

- Similar but not same!
  - Estimation techniques can be similar
  - Nature of Effort estimation is different
  - Models can be customized for Test Projects
  - Overheads and factors are different

- Test Projects have to consider these
  - Analysis of test types
  - Factors for estimation are based on Development environment and test environment
  - Demands on Skill, expertise and automation
  - Technology and infrastructure requirements
  - Domain knowledge
What is Important?

- Parameters to consider
  - Estimation methods
    - Can we derive from software estimation techniques
  - Framework
    - Is there a framework that can be built?
  - Knowledge and know-how?
    - Past Experience and know-how is crucial
    - Can we learn from the past?
    - Intelligence (metrics and data) from Projects is very important
  - Team
    - Teams play a crucial role
    - They are the backbone of the execution
    - Who else can predict accurately but them
    - Estimation is not a individual effort
    - It is a team effort!
Future Forward

- Learn from the past
  - Put metrics to best use
- Build a estimation model for testing projects
  - Use existing techniques to interpolate for testing projects
- Process Maturity
  - Build a model for quality of testing
  - Build reusable assets
  - Leverage existing maturity models of the organization
- People skill and talent pool
  - Incubate new talent
  - Identify methods and techniques
Estimation Techniques for Testing Projects

Topics
- Principles of Estimation
- Black Box
- White Box
- Performance Testing
- Stress Testing
- Other Testing
- Influencing Factors
- Team Dynamics
Principles of Estimation

- Principle of Estimation for testing projects shall be
  - Based on
    - Software requirements
    - Previous projects
    - Metrics
  - Estimation shall
    - Never forget the past
    - Be recorded
    - Be supported by tools
    - Be always verified
    - Consider automation needs
    - Consider people skills
  - Other Inputs to consider
    - # of test cases/scenarios,
    - # test cases per scenario,
    - # of builds (for regression)
Black Box

- Pointers for Estimation
  - Business Case
  - Function points
  - Object points
  - Requirement Specifications
  - Development Time
  - Estimate versus Actual
  - Automation
  - Test Bed Conditions
  - Technology requirements
  - Test Reporting
  - Skills
Business Case

- Business cases
  - Normal Scenario
    - Test prescribed functionality
    - Complete Details are available
  - Domain specific
    - Different types of domains require different calculations
    - Sometimes a combination is to be used
    - Not all testing can be completed
    - Scope of testing could be restricted based on the domain
    - Certain assumptions would have to be made
- Challenges – Abnormal Scenario
  - Details are not available
  - Specifications are incomplete!
Function Points

- **Function points**
  - Measure the size of a software application or project based on the number and complexity of user functions requested or delivered.
  - The calculation is objective.
  - Function points are independent of the technology.
  - The calculation can be used early in the development cycle.
  - Productivity history on completed projects with similar characteristics can be used.
  - The measurement is meaningful to non-technical users.
Function Points

- **Function points and Testing**
  - Can be used to estimate test effort
  - Each function point can help in developing relevant test cases
  - Nature of Testing (Black Box, White Box) decide the test cases and their complexities
  - Function point complexity mapped with test case complexity help in deriving test estimation
  - Are a good starting point
  - Combined with Requirement Specifications, Design and Analysis documents, help in arriving to near to actual test effort
  - Could be the best starting point for “Test Case development”
Object Points

- Object Points
  - Object points are an alternative function-related measure to function points when 4GLs or similar languages are used for development.
  - The number of raw objects are estimated, the complexity of each object is estimated, and the weighted total (Object-Point count) is computed.
  - Have nothing to do with objects in Object Oriented Programming.
Object Points

- **Object Points and Testing**
  - Can be used to estimate test effort
  - Each object point can help in developing relevant set of test cases or test suites
  - Re-usability, modularity can be built as against the FP based approach
  - Nature of Testing (Black Box, White Box) decide the test cases and their complexities
  - Object points are most often than not used as indicators of estimation
  - Accuracy on estimation is an iterative process
  - Ideally suited if you have more than one testing project in the same domain
Requirements Specifications

- Requirements Specifications
  - Depict the functionality of the system
  - Details the various components and their desired functionality
  - A good reference point to build test cases for both test to pass and test to fail
  - Test requirements are a good starting point if available
  - A well documented RS will have a good structure and a mature traceability
  - Could be the basis for estimation and project planning of Test Projects
Development Time

- Development of the System Under Test (SUT)
  - Time Taken
    - Time taken to develop the SUT is key
    - Indicates the trend on overrun, complexity
    - If actual time is available, it validates the estimation further
  - Interviews
    - If possible with development team members will provide indicators
  - Domain expertise
    - Prior experience of Testing team and Development Team – a great Plus
    - Past experience in testing similar projects
  - What to focus on – Test to pass or Test to fail?
    - Indicator that decides estimation largely
Estimate versus Actual

- Estimate versus Actual
  - Actual effort spent is a key indicator
    - If each RS item can indicate the time taken, and time allocated, variance can be effectively used
    - More about this in worksheet discussion
  - Actual metrics can provide indications
    - Complexity can be arrived at
    - Technology obstacles can be previewed
    - Tricky and Sticky code portions and implementations can be exposed
    - Nature of effort required to break can be judged
Automation

- **Automation**
  - A key indicator in estimating
  - Automation helps in regression
  - But it always takes more time to build automation suites
  - Trade off – Automation to Individual non-repeatable tests
  - Avoid automating when not deemed fit
  - Incorporate estimations for automation based on repeatability, reusability factors
  - Test Harness creation, maintenance
    - Key to identifying estimates
    - One time effort, but useless if it has no repeatability
    - If mandated, accommodate estimate to ensure that the overall estimate is not hurt
Test Bed Conditions

- Complimentary to Automation
  - Test bed needs more inputs if automation is involved
  - Automation decides the nature of test bed creation
  - Test Bed creation estimates have to consider
    - Initial and repeatable costs
    - Environmental considerations
    - Remote testing
    - Testing on live environments
    - Technology considerations
    - Hardware and software pre-requisites
    - Time to execute
Technology requirements

- Technology challenges are always new
  - New technology, old domain has hidden challenges
  - Support from Development teams crucial for estimation
- Challenges
  - Training time
  - Learning curve
  - Availability of experienced hands
  - Technical know-how is a prime factor
Test Reporting

- Test Reporting is a major activity
  - Estimates usually do not account for reporting
  - Reporting needs are based on process maturity and methodologies
  - Involve your QA Team to identify reporting needs and evolve estimations
  - Building new reports or Bug DB involves development effort
  - Keep the reporting simple if feasible
  - Use ‘Expert Judgment’ extensively
Skills

- Build your team skills continuously
  - Keep hiring the best minds
  - Allow good transition from projects
  - Build Domain expertise and prepare your team for the same
- Your estimate should allow time for skill building
  - Delivering all the time is not fun
  - Learning, delivering and re-learning is the way forward
  - Keep the human factor on top-priority
- Use an updated matrix of skills
  - Use an matrix to map your teams skills
  - Update the matrix regularly
  - Always Interpolate your estimate with skill set matrix
  - Build your own matrix that is suitable and “Keep it simple”
Using a Matrix

- The skill matrix can contain the following
  - Experience
    - Work Experience, Roles, Domains worked on
  - Skills
    - Soft skills
    - Testing skills on standards, process, SDLC, Testing maturity
    - Technology
    - Execution skills on Manual testing, automated testing, script building, automated test execution
    - Automation tools expertise, functional, structural and others
  - Process Maturity
    - Testing skills on planning, documentation, bug isolation, tracking
    - Design and development skills on black-box, white-box, static, dynamic testing, load testing, stress testing, etc
    - Configuration management knowledge
Black box - Summary

Summary

- Use a combination of Function/Object points and Requirement Specifications
- Arrive at test cases and the relevant test suites
- Find about development time estimated and actual
- Interview development team to identify critical paths to test
- Understand automation strategy and build plans for test bed creation/maintenance
- Test harness creation/maintenance should be accounted for
- Finalize your test results reporting techniques and account for the same in your estimate
- Review your skill matrix and map it to the estimate
White box

- Pointers for Estimation
  - Business Case
  - Lines of Code
  - Types of white box testing
    - Structural
    - Code Review
    - Coding Standards
  - Language factors
  - Technology considerations
  - Skills
  - Test Reporting
  - Automation
Business Case

- Business cases
  - Scenario for White Box Testing
  - Adherence to Standards
  - Too buggy code
  - Contractual requirement
  - Legacy code
  - Reverse or Re-Engineering
Lines of Code

- A major contributor
  - Lines of code is a decisive factor for estimation

- Other contributors
  - Complexity, requirement
  - Adherence to Standards
  - Structure of the code
  - Technology consideration
  - Programming Language
Types of white box testing

- Structural
  - Content
  - Reusability
  - Components
  - Maintainability
- Code Review
  - Walkthroughs
  - Discussions
  - Peer Review
- Coding Standards
  - Regulatory Standards
  - Internal Standards
  - Rules set by QA/Process teams
Language factors

- Programming Language
  - Key factors to look for
    - Complexity
    - Rules
    - Adherence to specifications
    - Support of various Standards
    - A key indicator for building the estimation based on skills available
    - Conventional, 3GL, 4GL and machine specific use of language
  - Object oriented languages
    - Increases complexity
    - Component based indices are to be verified
Technology considerations

- Technology
  - Plays a crucial role
  - Identifies what can be tested
  - Non Testable portions or grey areas should be investigated and documented

- Challenges
  - Training time
  - Learning curve
  - Availability of experienced hands
  - Consider using technology experts if skill set is not available internally
Skills

- Different needs from Black box Testing Skills
  - Usually developers with testing prowess is a safer bet
  - Programming language expertise is a must.
  - Good exposure to Software Engineering, metrics, quality indices are absolutely essential
  - Excellent analytical skills to avoid common pitfalls of misinterpreting code.
  - Good knowledge about standards
    - Especially true for embedded/ Mission critical projects
    - Aware of standards and its importance (E.g. DO178B)
Test Reporting

- Test Reporting is a major activity
  - Identify methods of reporting white box testing results
  - Adhere to standards of reporting, if required
    - Very critical when testing against standards
  - Identify critical estimates for building
    - Special reports
    - Reporting templates
    - Comprehensive detailing of white box errors
Automation

- Automation
  - Proven to be at least 10 times more effective
  - Statistical analysis can be much faster
  - Reporting is usually much faster and effective
  - Can reduce the need for
    - Variety of language skills
    - Skilled individuals with knowledge of Adherence to standards
    - Reporting mechanisms and templates
White Box - Summary

- Lines of Code plays an important role
- Programming Language is a key indicator
- Complexity increases when the testing types are more than one
- Skill demands are different from Black Box Testing
- Test results reporting is different
- Adherence to Standards and rules are of prime importance
- Automation reduces time
Performance Testing

- Pointers for Estimation
  - Business case
  - Domain
  - Nature of end users/usage patterns
  - Test Case development is an ongoing process
  - Framework based approach is best advisable
  - Use of automation tools an important factor
  - Automation frameworks provide a lot of estimation built in
  - Technology challenges are high
  - Test Bed Requirements are high
  - Test Data requirements are high
  - Preamble and post amble conditions are very stringent and extra focus is required
Stress Testing

- Pointers for Estimation
  - Business Case
  - Estimation is very similar to performance testing
  - Domain is crucial
  - Nature of end users/usage patterns
  - Critical tests take precedence
  - Test Data requirements are high
  - Technology challenges are high
  - Test Bed Requirements are high
  - Repeatability is a key issue to handle
Other Testing

“OPEN HOUSE” discussion
Influencing Factors

- High Complexity
- Stakeholders are many
- External Dependencies
- Lacking Skills in Domain, Technology
- Have to build new tools/frameworks
- Experimenting…
- Requirement of Hardware/Other infrastructure
- Test Data
- Legal contractual obligations
Team Dynamics

- People, People and People
  - The ones who make Good test Engineers
    - Anticipating the worst – Attitude hard to get, but should strive to find such attitude
    - Should be creative, testing is a creative work, however destructive the approaches are!
    - Curiosity is important, but should not kill the cat!
      - Idea is to find bugs, isolate and report them
    - Skill set mix, is quite important
      - Technical skills, inter-personal skills, soft skills, communication etc, all play a crucial role
    - Domain Experience, would be crucial
  - The team is likely to dilute, so keep the search on all the time
Worksheets

- Worksheets to use
  - Test Estimation Worksheet
  - Other Supplemental Worksheets
Test Estimation Worksheet

- Assimilates inputs from various estimation models to provide
  - Black Box testing estimation
  - White Box testing estimation
  - A basic building block
  - A Collaborate platform to build a comprehensive worksheet
Other Supplemental Worksheets

- Skill Matrix
- Test Bed Checklist
- Technology challenge checklist
- Domain expertise and past project inputs
Summary

- Use existing models to suit your test project requirements
- Estimations could be imprecise – but it’s ok!
- Wrong Estimations should be controllable!
- Re correct your estimate, if you have made a mistake – but do it early
- Collecting metrics and creating ‘Intelligence’ is crucial for success in the future
- Prepare for changes and challenges on a continuous basis
- Start Early and Stay focused
Summary

- Make your team understand dynamics of Test projects – all the time
- Allow team participation wherever possible
- Collaborate, retrospect and co-operate
- Experiment only on a small scale
- Build contingency and risk plans – but keep them trimmed
- Gain Confidence of management with methodologies and processes
- Put your team first and not yourself – it is better in the long run
Acknowledgments

Team – QSIT
Team – SAMGRAM, Chennai
Team – RESOLVERS, Bangalore

References:

Software Engineering
Ian Sommerville
Metrics and Models in Software Engineering
Stephen H Kan
Critical Testing Process
Rex Black

And all the “Software Testing Minds” around the Globe