Number Six Software

Project Sizing and Estimation

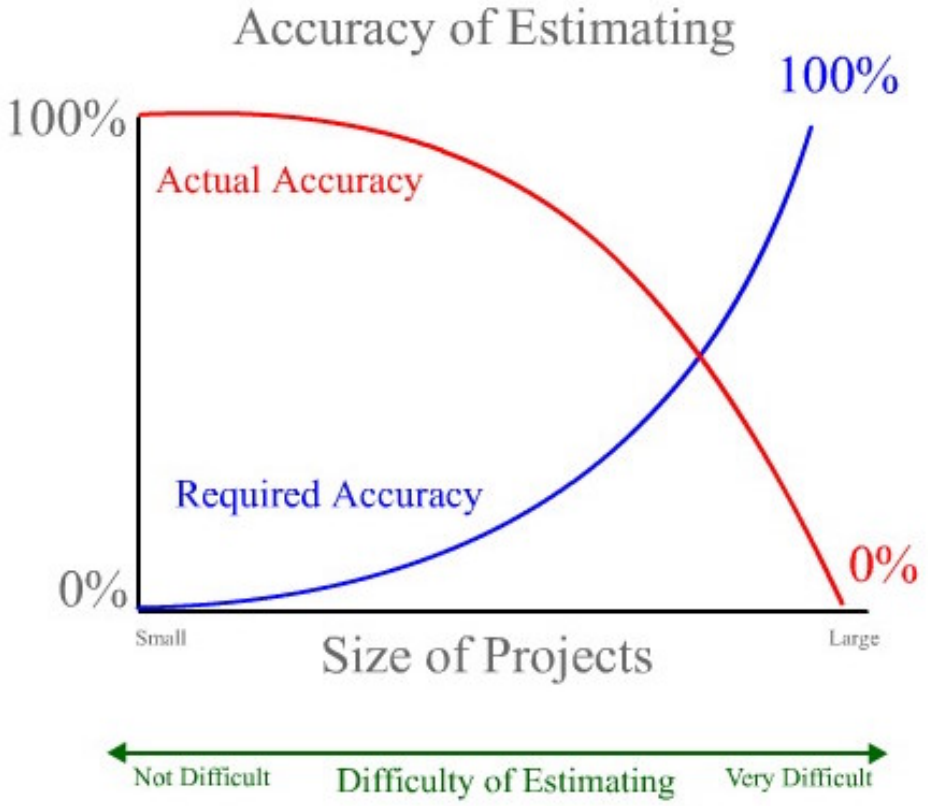
February 9, 2005

About Number Six

- Leading provider of expertise in developing, executing, and teaching the key elements of successful software development
- Full line of services in evolving software development using Practice Team approach
- Founded in 1994
- Locations in Arlington VA, Indianapolis IN, and Cincinnati OH
- Over 125 employees highly focused on making our customers successful

Estimating Software Development

- The inherent problem with estimating is that small projects are easy to estimate, but the required accuracy is not important – large projects are very difficult to estimate, but the required accuracy is very important



Why Estimate Software?

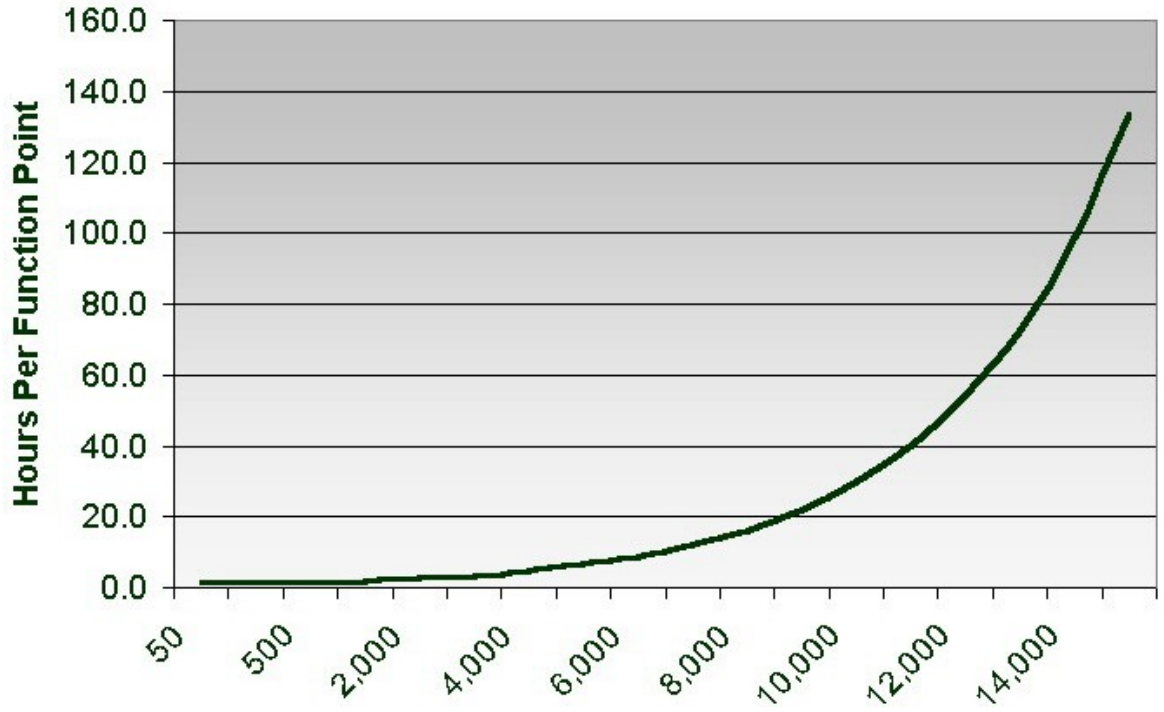
- Different requirements from estimation
  - Project Inception / Business Case
  - Once Functional Requirements have been identified
  - After requirements have been elaborated
  - Conclusion of Elaboration activities
  - Post construction
  - Project conclusion

Characteristics of a Good Estimate

- Explainable
  - Should have a sufficient amount of granularity so that the estimate can be explained
- Based upon historical data
- List of Assumptions
- Revisable
  - As more is known about the project, revisions can be made to provide a more accurate estimate

Impacts to Project Estimation

- Size
  - The greatest impact on productivity, and therefore project estimation, is the size of the project



David Longstreet, Estimating Software Development Effort

Impacts to Project Estimation

- Definition of functional requirements
- Personnel skill sets
  - .5 to 5 hours per function point on small projects
  - 20-60 hours per function point on larger projects
- Management skills
  - Management skills flatten the productivity curve
- Technology
  - Components of technology produce variation in rates of productivity
    - A Cobol programmer may be able to deliver 10 function points per man-month – a VB developer typically can deliver 40 function points per man-month

Estimating Software Development

Need to accurately depict the size of software based on well-defined requirements using a repository of accurate historical data, taking into account the personnel and management skills on a targeted technology platform

Components of Estimating

- Well-defined requirements
  - Formal requirement management approach
- Accurately depict the size of software
  - KLOC – Lines of Code
  - Function point analysis
- Repository of accurate historical data, personnel expertise, and management skills
  - Knowledge Plan estimation tool

What is a Function Point?

Function points measure software size by quantifying the functionality provided to the user based solely on logical design and functional specifications. With this in mind, the objectives of FP are to:

- Measure functionality that the user requests and receives
- Measure software development and maintenance independently of the technology used for implementation
- Provide a normalizing measure across projects and organizations

What is a Function Point?

- Allen Albright of IBM published the function point metric in 1979
  - Software could be sized by evaluating the external transactions processed by an application or subsystem and the databases that were used
- Further enhanced in 1984 to include individual complexities and system characteristics
- International Function Point User Group established in 1986
  - Provides standardization and promulgation of the metric
  - More than 1,400 current member organizations

Function Point Analysis

- Function point analysis is based on the theory that the functions of an application are the best measurement of a software application's size
- Function point analysis breaks systems into smaller components to facilitate better understanding of the system functions – similar to use cases

Why use Function Points?

- You can’t manage internally what you don’t measure
- Approximately 40% of projects fail due to lack of management control
- Measurement provides a tool to communicate the size of change requests
- To understand and improve your software development processes

Why use Function Points?

- Map consistently with use cases
  - FP's are used to size software from a requirements perspective
  - UC's are used to develop requirements
- Large historical basis for comparison
  - Easy to compare UC methods versus other methods
  - Easy to compare past project estimates to current project estimates
- Easy to change project estimates and explain variances – the more you know the better your estimates

Identifying Function Points

NUMBERSIX

- Identify logical data groups used within the application
  - Internal Logical Files (ILF) - maintained
  - External Interface Files (EIF) - referenced
- Identify transactions used within the application to provide external inputs, outputs and inquiries
  - External Input (EI)
  - External Output (EO)
  - External Queries (EQ)

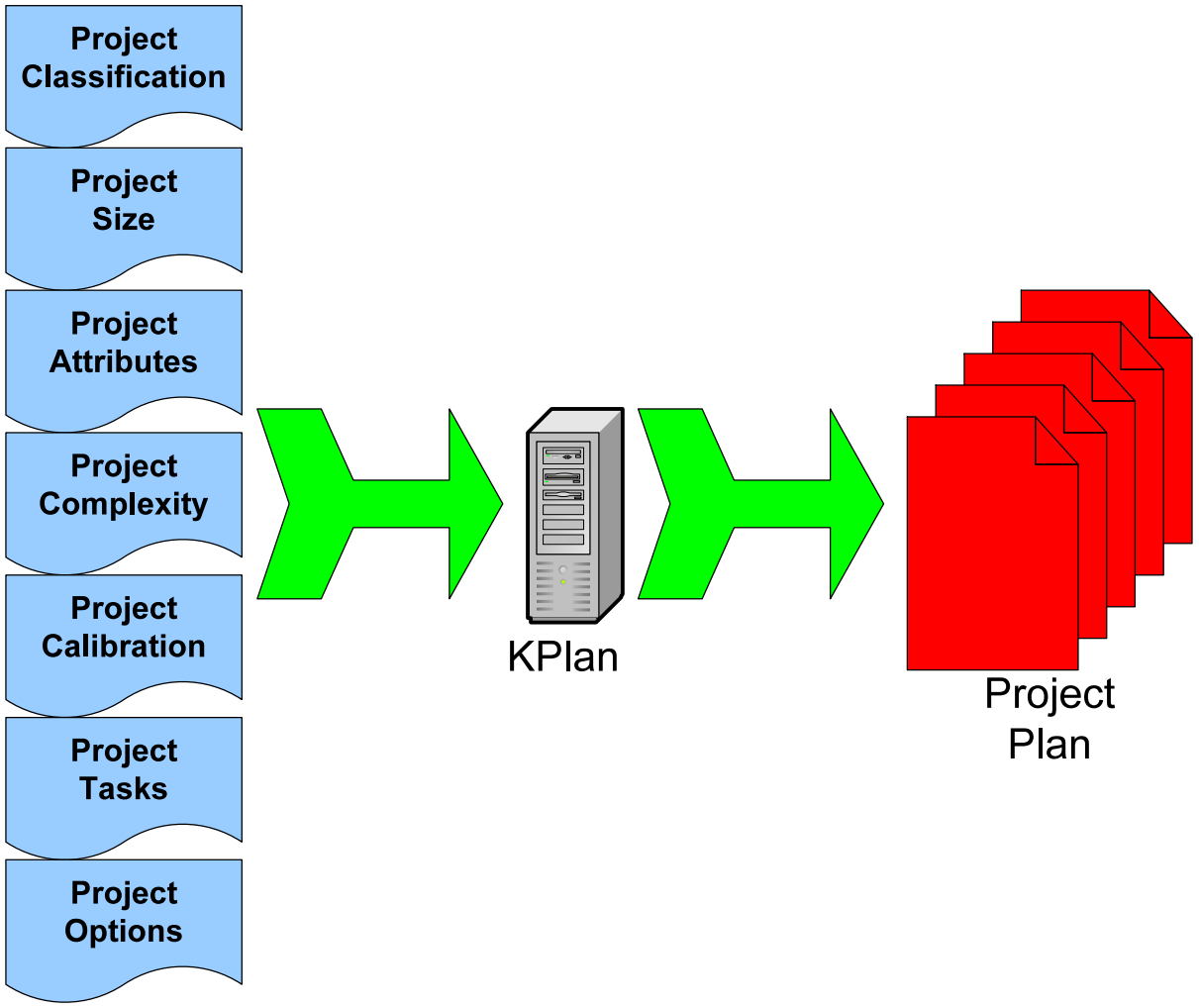
Calculating Function Points

- Each logical data group is rated as High, Medium, or Low
  - Values are calculated using standard matrices
  - Entered into function point collection spreadsheet
- Factors applied to accommodate system characteristics
- Summary of function point analysis provides adjusted count of application size

Using the Function Point Estimate

- Function point estimate provides the “size” metric
- Use the size metric against historical collection of project data to derive estimate
- Knowledge Plan (KPLan) provides this repository of historical data
  - Provides the “past experience” to which an estimate can be compared
- Process provides consistency in uncovering all necessary and important data that describes the project
  - Removes subjectivity that is normally applied to estimating projects

Knowledge Plan Estimation



Knowledge Plan Estimation

- KPLan includes a collection of over 10,000 projects to compare against
- Minimizes variance in estimates completed by different resources, and at different project milestones
- Three important phases
  - Classification of the project
  - Sizing of the project
  - Anticipating the productivity in the development environment

SPR Knowledge PLAN

Summary

- Apply structure and process to sizing and estimating
  - Use of recognized methods / standards
- Provide accurate and consistent results regardless of resource and phase
- Standard method to assess change
- Comparison against extensive repository of historical data

Estimate early and often!