Web Development: Estimating Quick-to-Market Software

Donald J. Reifer

15th International Forum on COCOMO and Software Estimation

Setting the Stage

- Business and government are rushing to web
- Many organizations are embracing quick-to-market paradigms (death march projects, etc.)
- These changes result in new way for applications to be engineered

Web Versus Traditional Projects

Characteristics	Traditional	Web
Primary aim	Build products at minimum cost	Bring products to market quickly
Typical size	Medium to large (10 – 100+ engineers)	Small (3-10 engineers)
Timeline	12-18 months	3-6 months
Technology used	OOT, CASE tools, generators, C++, etc.	CBSE, frameworks, Java, multi-media, etc.
Process	CMM-based	Ad hoc, death marches

Reasons for Death Marches

- Startup mentality of fledging, entrepreneurial companies
- Naïve promises made by marketing, senior executives, inexperienced project leaders
- The "Marine Corp" mindset – real programmers don't need sleep (feed them pizza and keep them coding)
- Intense pressure put on for quick return by investors
- Heat placed on firm due to globalization of marketplace
- Many unplanned and unexpected crises (turnover, etc.)

Source: Yourdon, ***Death March***, Prentice-Hall, 1997

More on Web Versus Traditional

Characteristics	Traditional	Web
Products	Code-based systems, done in-house, mostly new, many external interfaces, often complex	Object-based systems, multi-media, done out-house, many reusable parts, few external interfaces, often simple
Development staff	Professional software engineers	Graphics designers, software engineers, etc.
Estimating technology	Size: SLOC or fp Resources: models or WBS estimate	Size: ??? ***(web objects)*** Resources: ad hoc or via WBS estimate

Web-Based Estimating Challenges

Characteristics	Traditional	Web
Estimating process	Most use analogy and past experience	Most use WBS (inputs from developers)
Size estimation	Requirements drive estimates • SLOC or function points used • Legacy, reuse, etc. handled as % new	Applications are built using frameworks or templates (90% solution) • Web objects proposed (applets, video, etc.) • No agreed to metric
Schedule estimation	Most estimated using cube root relationship	Analysis show estimates too long (square root?)

Web-Based Estimating Challenges

Characteristics	Traditional	Web
Effort estimation	Estimated using regression or via knowledge base • Modified by cost drivers • Allocated via rules or Rayleigh curves	Estimated by breaking the job down into tasks and having responsible engineer estimate what is needed to do job • Little history • No knowledge base
Calibration	Use past experience	Based on folklore
"What-if" analysis	Quantitative	Qualitative

New Size Metrics

Traditional

- Source lines of code
  - Non-visual, no motion, etc.
- Function points
  - Conventions don't exist
- Function point extensions (3D, predictive object points, etc.)
  - Requirements driven, often OO-based, non-visual
- UML metrics
  - Same as extensions

Non-Traditional

- Multi-media files
  - **Graphics - Text**
  - **Audio - Motion**
  - **Pictures (JPEG, etc.)**
- Web components
  - **Applets, Agents, etc.**
- Fine-grained building blocks
  - **DCOM, CORBA, etc.**
- Coarse-grained components
  - **Shopping carts, etc.**

Time for Some Heresy

Halstead's Equation for Volume

$$V = N \log_2 (n) = (N_1 + N_2) \log_2 (n_1 + n_2)$$

Where: N = number of occurrences of Operands/Operators
n = number of distinct Operands/Operators
N_1 = total occurrences of Operand estimator
N_2 = total occurrences of Operator estimator
n_1 = number of unique Operands estimator
n_2 = number of unique Operator estimator
V = volume of work involved represented as we objects

Web-Based Size Predictors

Web Object Predictors	Example Operands	Example Operators
# of building blocks	Widgets, fined-grained components (ActiveX, DCOM, OLE, etc.), etc.	Create, apply, call, dispatch, interface, terminate, etc.
# of COTS components	COTS programs, library routines, web objects (cart), etc.	Transform, access, bind, generate, interface, etc.
# of multi-media files	Text, video, audio, etc. (not graphics files)	Create, cut, paste, clear, edit, etc.
# of application or object points (or others proposed)	# server tables, # client tables, # states, # entities, attributes, etc.	Transform, access, modify, instantiate, generate, etc.

More Web-Based Predictors

Web Object Predictors	Example Operands	Example Operators
# of web components	Applets, agents, guards, etc.	Create, schedule, dispatch, etc.
# of xml, sgml, html and query lines	Lines including links to data attributes	Create, call, browse, link, traversal, etc.
# of graphics files	Templates, pictures, images, etc.	Apply, align, import, export, insert, etc.
# of scripts (visual language, audio, motion, etc.)	Macros, containers, etc.	Create, store, edit, distribute, serialize, generate, etc.

Web Object Calculation Worksheet

Web Object Predictors	Low	Medium	High
# of building blocks	1	2	4
# of components	2	4	6
# of multi-media files	1	2	4
# of application or object points	*	*	*
# of web components	2	4	6
# of xml, sgml, html & query lines	3	5	8
# of graphics files	2	4	6
# of scripts	1	2	3
Other	2	4	6

*** Assume weights have already been applied**

Interesting Findings

- Size of application is extremely sensitive to counting conventions
  - Must use one set of conventions consistently
  - We use IFPUG and weighted methods per class
- Fit data using web objects, not SLOC or other metric
- Weighted goodness of fit significantly better than for unweighted predictors
  - Based upon 46 completed projects whose size ranged from 20 to 100 web objects
- Currently defining weightings more precisely

Estimation via COCOMO II

- Effort estimation via Early Design model
  - Had co-linearity in the data which was noisy
- Simplified the effort estimation model further to eliminate co-linearity/fit the data
  - Resulting model has eight drivers and fixed power laws based upon four types of web applications researched
- Schedule estimation via Early Design model
  - Estimate was way off so used square instead of cube root relationship (based on scaling rules)

WEBMO – Web Estimation Model

$$\text{Effort} = A \prod_{i=1}^{8} C_i \,(\text{Size})^{P1} \qquad \text{Duration} = B(\text{Effort})^{P2}$$

Where: A = effort coefficient
B = duration coefficient
C_i = cost drivers
P1 = effort power law
P2 = duration power law

Cost Drivers

- Kept Early Design model drivers:
  - **RCPX** – Product Reliability & Complexity
    - RELY, DATA, CPLX, DOCU
  - **PDIF** – Platform Difficulty
    - TIME, STOR, PVOL
  - **PERS** – Personnel capability
    - ACAP, PCAP, PCON
  - **PREX** – Personnel experience
    - APEX, PLEX, LTEX
  - **FCIL** – Facilities
    - TOOL, SITE
  - **SCED** – Required development schedule
- Eliminated one driver
  - **RUSE – not significant**
- Could not calibrate with scale factors; used two new cost drivers to handle
  - **TEAM** - Teamwork
  - **PEFF –** Process Efficiency

WEBMO Parameters/Power Laws

	A	B	P1	P2
Web based electronic commerce	**2.3**	**2.0**	**1.05**	*
Financial/trading application	**2.7**	**2.2**	**1.05**	*
B-to-B applications	**2.0**	**1.5**	**1.00**	*
Web based information utilities	**2.1**	**2.0**	**1.00**	*

* Either 0.5 or 0.33 based upon scaling (> 40 web objects)

Driver Rating Scales

Driver	VL	L	N	H	VH
RCPX	- Client side - No distribution - Invocation - Simple math - Simple I/O - Limited data - Reliability not a factor	- Client/server - Ltd distribution - Adaptation - Standard math - File management - Some files - Easy to recover from losses	- Client/server - Fully distributed - Integration - Statistics - DBMS - Databases - Moderate recovery goals	- Configurable web - Wide distribution - Synchronization - Math intensive - Distributed DB - Virtual database - High financial loss due to error	- Seamless web - Full distribution - Collaboration - Soft real-time - Persistent DB - Virtual database - Errors cause risk to human life
PDIF	-Rare changes to platform - Speedy net - Best possible connectivity - No computer resource limitations	-Few changes to platform -Fast net service - Rare lose of connectivity -Few computer resource limitations	-Platform stable - Acceptable net performance -Acceptable connectivity -Must watch use of computer resources	-Frequent changes to platform - Slow network performance - Poor connectivity - Lack of computer resources causing problems	-Platform not stable -Unacceptable net performance - Unacceptable connectivity -Timing and storage impacts

Driver Rating Scales

Driver	VL	L	N	H	VH
PERS	- 15th percentile - Major delays due to turnover and attrition	- 35th percentile - Minor delays due to turnover and attrition	- 55th percentile - Few delays due to turnover and attrition	- 75th percentile - Infrequent delays due to turnover and attrition	- 90th percentile - No delays due to turnover and attrition
PREX	- ≤ 2 months average applications, platform, language and tool experience	-≤ 6 months average applications, platform, language and tool experience	- ≤ 1 year average applications, platform, language and tool experience	- ≤ 3 years average applications, platform, language and tool experience	- ≤ 6 years average applications, platform, language and tool experience

Driver Rating Scales

Driver	VL	L	N	H	VH
FCIL	- International - Phone/fax - Ad hoc methods - Language tools - Basically no collaboration	- Multi-site - Phone/email - Phase dependent methods - Basic CASE - Limited collaboration	- One site - LAN - Life cycle methods - Tools support methods - Integrated product teams	- Same building - WAN - Integrated methods - Integrated toolset - Collaborative teams	- Co-located - Broadband - State-of-the-art methods - Integrated toolset that supports collaboration
SCED	- Must shorten -75 percent of nominal	-Must shorten -85 percent of nominal	- Keep as is - Nominal	- Can relax -130 percent of nominal	- Can extend -160 percent of nominal
REUSE	Not used	Not used	Note used	Not used	Not used

Driver Rating Scales

Driver	VL	L	N	H	VH
TEAM	- No shared vision - Stakeholders do not work to meet each others objectives - Teamwork limited	- Little shared vision - Stakeholders talk and build some respect each other's goals - Some teamwork	- Some shared vision - Stakeholders pull together and work towards joint goals - Basic teamwork	- Considerable shared vision - Stakeholders respect each others goals and collaborate on the effort - Integrated teams	- Extensive shared vision - Stakeholders pull together from day 1 and mount a goal-directed effort - Seamless teams
PEFF	- Totally ad hoc, confused process - Reliance on hero's to get job done	- Project-based processes (little sharing with others) - Reliance on management leadership to meet goals	- Streamlined process tailored for fast-paced jobs and teams - Reliance on process for guidance	- Efficient process matched to getting the job done - Process is the way engineers do their work	- Effective process used to generate quality products on-time & budget - Everyone uses and believes in the process

Cost Driver Values

Driver	VL	L	N	H	VH
RCPX	0.63	0.85	1.00	1.30	1.67
PDIF*	0.75	0.87	1.00	1.21	1.41
PERS	1.55	1.35	1.00	0.75	0.58
PREX	1.35	1.19	1.00	0.87	0.71
FCIL	1.35	1.13	1.00	0.85	0.68
SCED*	1.35	1.15	1.00	1.05	1.10
TEAM	1.45	1.31	1.00	0.75	0.62
PEFF	1.35	1.20	1.00	0.85	0.65

*** Significant differences from original model**

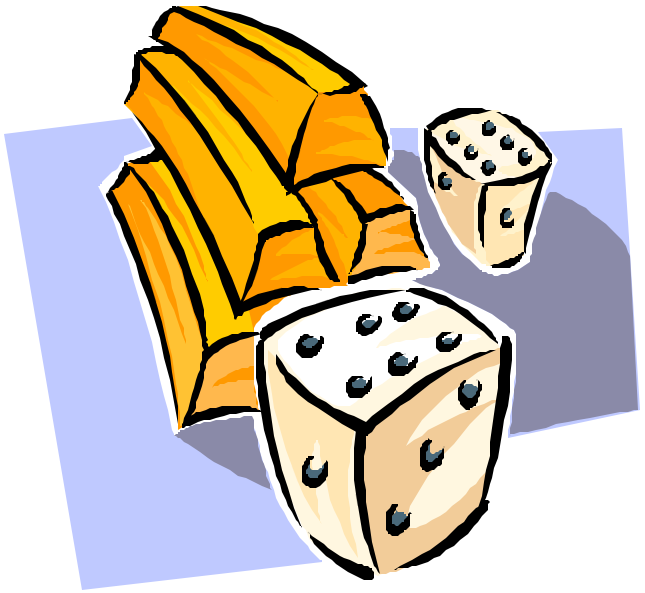
Current Status

- Goal is to help clients better estimate resources needed for web development
- In the process of learning, validating and extending the model
  - Current database has 46 completed projects (need more)
  - Using linear regression to fit the actuals to estimates
  - Data is noisy (no timecards, 80 hour weeks, etc.)
- This is not a USC sponsored project
  - Our intent is to develop a model for use in our consulting business
  - We are funded by several clients and have a proposal in to NSF for a grant

Next Steps

- Analyze more data
- Determine if we can use Early Design model or CORADMO out of the box
- Improve predictive accuracy
  - Current accuracy is estimates are within 30 percent of actuals 60 percent of the time
  - Better than winging it, but not good enough
- Have fun, help clients and make money
  - Reifer Consultants is a for profit firm

Final Thoughts



- The rush to the Internet represents a modern day gold rush
- Those who provision the prospectors will make the fortunes
- We want to provide them with tools they need to figure out what it will take for them to be successful

Acknowledgements

- USC Center for Software Engineering
  - Barry Boehm and the COCOMO team
- Galorath Inc.
  - Lee Fischman and Dan Galorath
- Marotz
  - Bill Roetzheim
- The Multimedia House of Quality
  - Adrian Cowderoy
- Those nameless individuals who worked hard to supply the data

