BEYOND THE CRYSTAL BALL:

PROTOTYPING STRATEGIES AND TOOLS FOR DETERMINING SOLID METRICS, SHRINKING SCHEDULES AND COSTS, AND IMPROVING QUALITY

Deborah L. Stone

Steven W. Villachica, Ph.D.

Presented at the April 2002 Annual Conference of the a International Society for Performance Improvement, Dallas, TX

Supplemental Materials

TABLE OF CONTENTS

Beyond the Crystal Ball

Beyond the Crystal Ball

Prototyping Strategies and Tools for Determining Solid Metrics, Shrinking Schedules and Costs, and Improving Quality

Deborah L. Stone
CEO and President

Steven W. Villachica, Ph. D.
Chief Learning Officer

www.dls.com

Paper presented at the 40th Annual International Performance Improvement Conference & Expo, Dallas, TX.

Pressures We All Feel

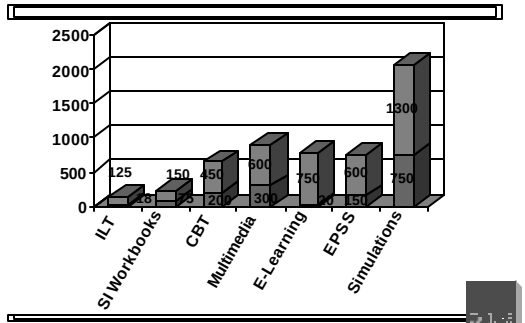
- Complex problems
- Unhappy end users
- Use of call centers
- Revision costs

- Tolerance for failure
- Product to market time
- Product life cycle

Agenda

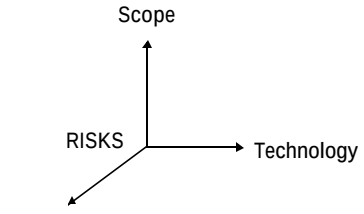
- ✓ Describe industry development metrics and their limitations
- Describe five proven strategies to see beyond the crystal ball
- Use the Project Metric Spreadsheet to compare estimated to actual metrics
- Describe the additional benefits associated with prototype-based development approaches

Development Metrics
Industry "Standards" Are Highly Variable



4

Factors Affecting Metrics
Why the "Standards" Are So Variable



—Adapted from Foshay (1994)

5

Given This Variability…
…You Need to Establish Your Own Metrics

- **Turn to similar past projects**
  - **Scope of work**
  - **Costs**
  - **Schedule**
  - **Reusable objects**
- **Extrapolate into the future based on rapid prototyping**

6

Agenda

- ✓ Describe industry development metrics and their limitations
- ✓ Describe five proven strategies to see beyond the crystal ball
- Use the Project Metric Spreadsheet to compare estimated to actual metrics
- Describe the additional benefits associated with prototype-based development approaches

7

Rapid Application Development (RAD)

A software development methodology designed to decrease development time while improving quality

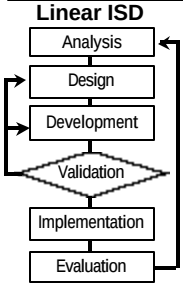
- James Martin (1991)

8

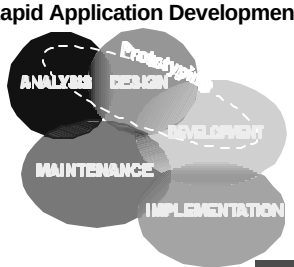
Linear ISD versus RAD

Two Methodologies

Linear ISD



Rapid Application Development

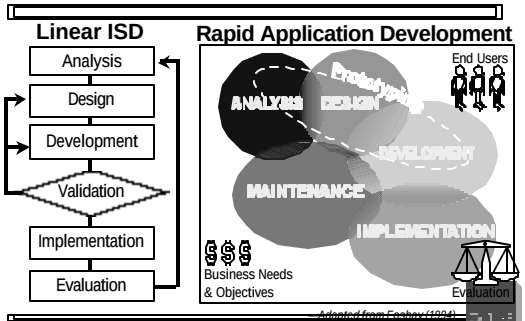


— Adapted from Foshay (1994)

9

Beyond the Crystal Ball

Linear ISD versus RAD
Two Methodologies



Adapted from Foshay (1994)

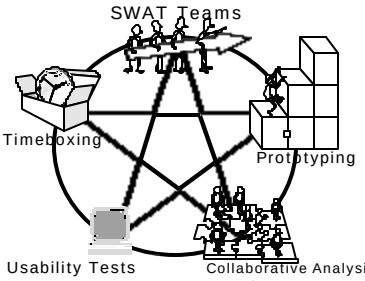
10

Linear ISD vs. RAD
When to use each

Linear ISD	RAD
■ Low amount of risk	■ High amount of risk
■ Known requirements	■ Fuzzy requirements
■ Known designs	■ Unknown designs
■ Known development processes	■ Unknown development processes
■ Trustworthy development metrics	■ Unknown Development metrics

11

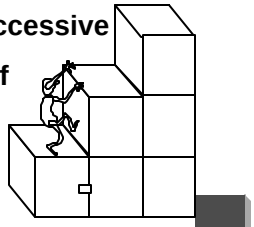
Five Proven RAD Strategies
Checks and Balances



12

Prototyping

An iterative design approach that produces successive approximations of the end product

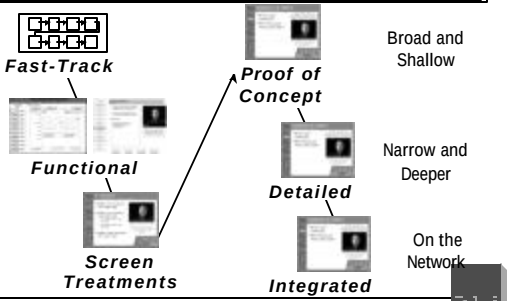


Why Prototype?

- Test the accuracy and completeness of performance requirements.
- Ensure the design of the intervention meets the performance requirements.
- Facilitate end-user "buy-in"
- Create an "assembly line" for the development effort.

Prototyping Process

A Step-by-Step Method



Project Metric Spreadsheet

Process

1. Specify process flow (page 12 of supplemental materials.)
2. Provide a description of each activity, as well as corresponding inputs, process, decisions, responsibility, and output.
3. Build the deliverable(s) and track actual completion times and variances
4. Conduct a post mortem
5. Modify and revise the project metric spreadsheet
6. Complete steps 1-5 for the next prototype or the development effort

16

Prototyping Details

1. Fast-Track Scenario
2. Functional Prototype
3. Screen Treatments
4. Proof of Concept
5. Detailed Prototype
6. Integrated Prototype

17

Agenda

- ✓ Describe industry development metrics and their limitations
- ✓ Describe five proven strategies to see beyond the crystal ball
- ✓ Use the Project Metric Spreadsheet to compare estimated to actual metrics
- ■ Describe the additional benefits associated with prototype-based development approaches

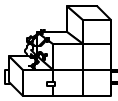
18

Project Metric Spreadsheet
Rules of Thumb

- Prototype what's representative and difficult
- Specify all processes
- Be sure to include a fudge factor in estimated times
- Allow team members to estimate and track their own completion times
- Specify quality standards for each output
- Use logbooks to track prototype completion, included expected and unexpected activities
- Concentrate on the time-sucking, black holes during the post mortem
- Reinforce cycle time shrinkage

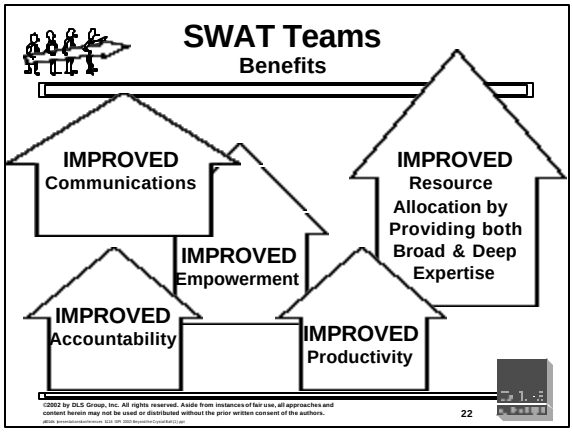
Agenda

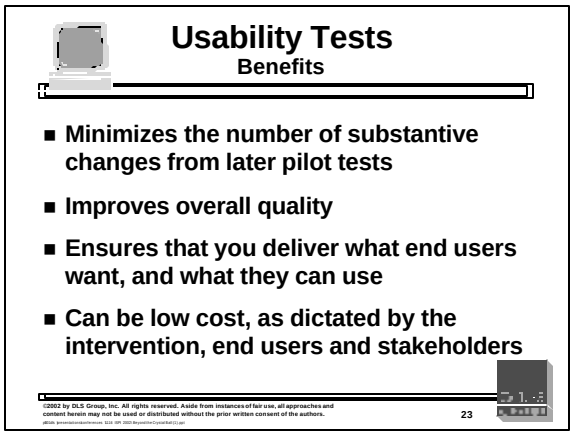
- ✓ Describe industry development metrics and their limitations
- ✓ Describe five proven strategies to see beyond the crystal ball
- ✓ Use the Project Metric Spreadsheet to compare estimated to actual metrics
- ✓ Describe the additional benefits associated with prototype-based development approaches

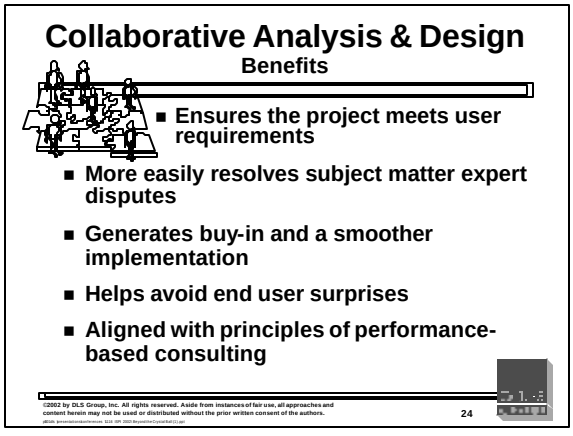


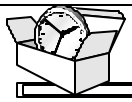
Prototyping
Benefits

- Provides accurate metrics
- Sets up a finely tuned assembly line for the development process
- Tests the accuracy and completeness of performance requirements
- Generates buy-in using user-accepted designs
- Minimizes changes later in the project when they are more expensive
- Delivers results quickly and inexpensively





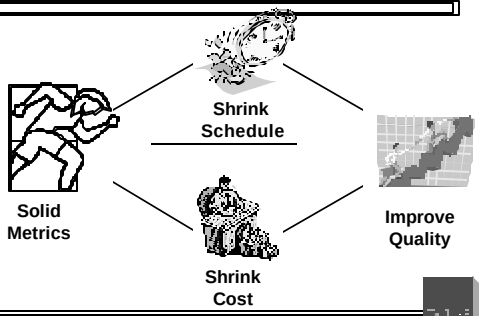




Timeboxing
Benefits

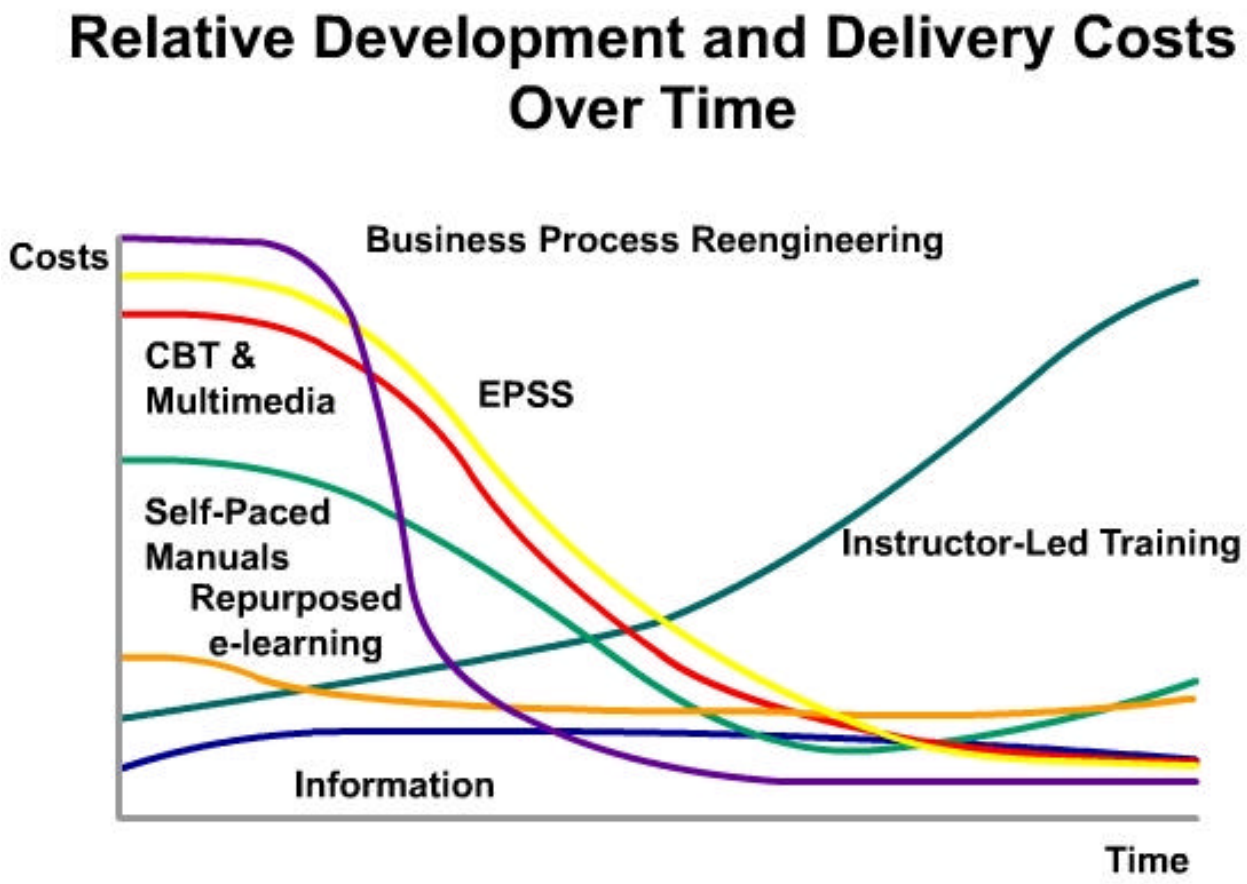
- Quickly produces completed materials
- Facilitates rapid revision to completed materials
- Controls scope and schedule creep sometimes associated with prototyping and collaborative analysis and design
- Adds to the project on subsequent revisions—if the business and performance needs still exists

Prototyping and RAD Strategies



QUESTIONS AND COMMENTS

Development Versus Delivery Costs

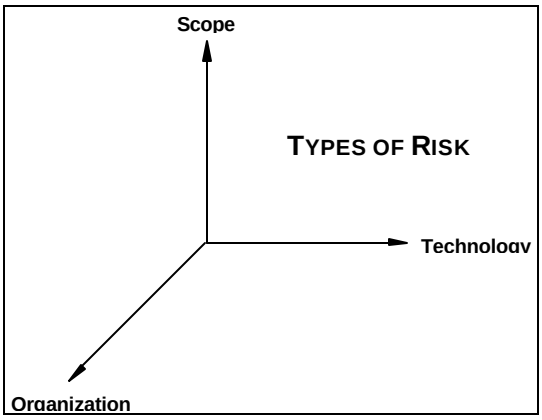


DEVELOPMENT RATIOS

Delivery Medium	Development Time to Create 1 User Contact Hour
• Instructor-led Training	18 - 125 hours[1]
• Self-Instructional Manuals	75 - 150 hours [1]
• Computer-Based Training	200 - 450 hours[1]
• Multimedia	300 - 600 hours[1]
• Web-based and E-learning	20 - 750 hours[1]
• Performance Support System	150 - 600 hours
	Development Time
• Hypertext Information Systems	2 - 5 hours per pop-on window
• Task-based Online Help/Cue Cards	3 - 5 hours per pop-on window
• Procedural Software Documentation (print-based)	$100 per page[1] 1 - 4 hours per page
• Policy/Procedure Documentation (print-based)[1]	$120 per page[1]
• Linear Video	$1,500 - $2,500 per running minute 3-10 hrs for videotaped lecture; 100 hours for broadcast quality
• Interactive Video	300 hours per hour
• Animation	$3,000 - $8,000 per minute
• Expert Systems	0.5 - 8 hours per rule

[1] Excludes costs for printing, packaging, and shipping.

FACTORS THAT AFFECT DEVELOPMENT RATIOS



ORGANIZATIONAL FACTORS

- Span of control and level of project sponsor.
- Dedication and relationship of client and supplier project managers.
- Quality of product (from both the customer's and supplier's perspective).
- Number of reviews and timelines for sign-off.
- Previous experience with the customer.
- Amount of end-user involvement in analysis, design, and implementation.
- Amount of available expertise in the subject matter.
- Quality of communications.
- Presence of hidden agendas.
- Expected amount of time spent not related to designing or revising the instructional materials. Some developers estimate that this typically consumes about 80 percent of project time.

SCOPE FACTORS

- Number of expected user contact hours.
- Cognitive performance requirements (conceptual, procedural, problem-solving).
- Complexity of content.
- Number, type, and complexity of components.
- Availability, quality, and accuracy of existing content.
- Complexity and frequency of interactions (performance requirements).
- Specificity of the performance requirements.
- Number and complexity of graphics, animation, and multimedia.
- Ease of use.
- Familiarity of target audience with medium.
- Quality of finished product requirements.
- Length of course.
- Degree of remediation (e.g., quizzes, selective module reviews).

TECHNOLOGY FACTORS

- Development and implementation platform, authoring and architecture / network environment.
- Distribution.
- Available bandwidth.
- Familiarity with development approach.
- Availability and expertise in specialized development tools, libraries, and templates.

- Experience of development team.
- Degree to which development team has worked together before.
- Rigor of the project management and change management processes.
- Development model the project team employs (traditional ADDIE's linear, "waterfall" approach versus Rapid Application Development (RAD) or other 4th generation ISD model).
- Availability of project management data describing a similar development effort.
- Availability of appropriate templates or toolsets.
- Need for specialized peripherals (e.g., touch screens, digitizers, robotics).

RAPID APPLICATION DEVELOPMENT FACTOIDS

Prototyping

- Organizations within the United States have increased their use of prototyping from approximately 30 percent in the early 1980s to 60 percent in the early 1990s. (Hardgrave and Wilson, 1994)
- Gordon and Bieman (1995) cite individual cases where prototyping reduced software development efforts by 45 percent and 70 percent.
- Bernstein and Appel cite an approximate 40 percent reduction in software development effort associated with prototyping.
- Martin (1983) reports that productivity improvements of more than 1,000 percent are not uncommon when using prototyping.
- A software revision requiring $1 to make during the analysis phase could cost between $1.50 - $6.00 to make during development and between $60-$100 to make after release. (Pressman, 1992)
- Pressman (1982) estimates that 80 percent of total software life-cycle costs occur during maintenance. Of these maintenance costs, Martin and McClure (1983) attribute 80 percent of them to unmet or unforeseen user requirements and 20 percent to bugs or reliability problems.
- Dissatisfied software end users often become "project terrorists" and will tell an average of 16 other people about his or her problems with a system. In contrast, each satisfied user will tell only eight people (Merlyn, 1995).
- Prototyping reportedly reduced one plastic manufacturer's development cycle by 60 percent. (Griffiths, 1993)

Collaborative Analysis and Design

- JRPs and JADs exploit the 80:20 rule (80 percent of value of the intervention can be achieved with 20 percent of the intervention) by specifying and designing the 20 percent. (Villachica and Stone, 1998)
- End users involved in JRPs and JADs will act as "project ambassadors," jump-starting the implementation effort early in the project. (Villachica and Stone, 1998)
- JAD cuts the elapsed time of requirements specification anywhere from 20 to 60 percent compared with traditional methods, and it cuts total effort by 20 to 60 percent at the same time (August, 1991).

Timeboxing

- An argument for using timeboxing is that 75 percent of a given intervention's functionality can be developed relatively quickly by using expertise, powerful tools, and archives of reusable materials. The next 15 percent of the intervention can take as long to develop as the first 75 percent. The remaining 10 may take as long to develop as the first 90 percent (Martin, 1990c).
- DuPont has used timeboxing since 1984 to maintain its 90-day software development cycle. After its initial use, DuPont has not rejected a single system developed with timeboxing (McConnell, 1996).

SWAT Teams

- DLS Group, Inc. (1994) programmed 11 hours of CBT in two weeks using a SWAT team strategy.

Usability Testing

- The top five educational software companies remaining in the market over the last 10-15 years all employ usability tests (Abelow, 1993).
- IBM indicated that smaller software projects saved $2 for every $1 spent on usability tests. Large software development projects save $100 for every $1 spent on usability tests. (Appleton, 1993)
- Karat (1992) describes usability test payback ratios of 1:2, 1:10, and 1:500.
- IDS Financial Services changed a single form as the result of a usability test, resulting in greater productivity and saving the company $116,000 annually in labor costs. (Reed, 1993)

- State Farm leveraged usability testing to reduce a course the company thought would require three days to a half day, saving roughly two days' pay for approximately 10,000 students. (Computerworld, 1999)
- Research has shown that incorporating user feedback is the single most critical factor in improving learning outcomes. (Weston, C. 1997)

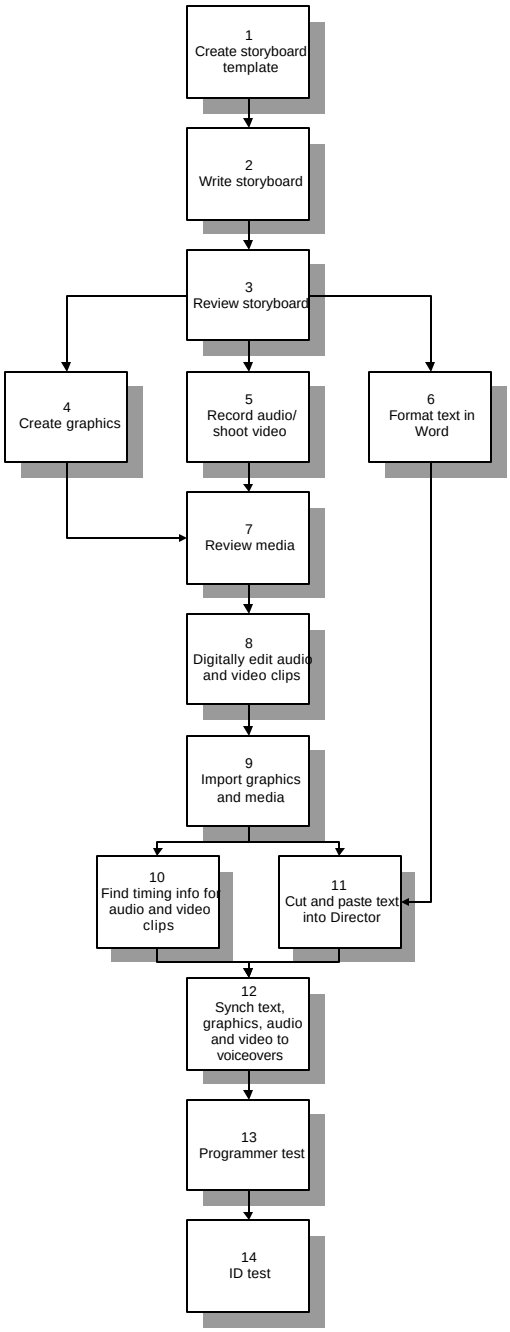
PROTOTYPING PROCESS

Phases	Input	Process	Decisions	Responsibility	Output
1. Fast-track scenario	• Performance requirements from Joint Requirement Planning (JRP) Workshop	• Convert performance requirements into a fast-track scenario • Draw the screens comprising the fast track on a whiteboard • Client review	• How will users interact with the intervention? • What screens do we need to represent these interactions? • Is the fast-track scenario consistent with the performance requirements? • Do the performance requirements need to be modified for accuracy and completeness?	• Programmer • Lead HPT • SME • Graphic artist	• Approved fast-track scenario
2. Functional Prototype	• Fast-track scenario • Performance requirements from Joint Requirement Planning (JRP) Workshop	• Create PowerPoint prototype • Client review	• Does the functional prototype meet the specified performance requirements? • Do the performance requirements need to be modified for accuracy and completeness?	• Lead HPT	• Approved functional prototype

Phases	Input	Process	Decisions	Responsibility	Output
3. Screen Treatments	• Approved functional prototype • Client identity materials	• Create Adobe Photoshop & Illustrator depictions of the main screens • Client review	• What should the user interface look like? • What graphic combinations will provide the best look and feel? • Do the screen treatments meet the performance requirements? • Do the performance requirements need to be modified for accuracy and completeness?	• Graphic artist • Lead HPT	• Approved screen treatments
4. Prototype Proof of Concept (broad and shallow)	• Functional prototype • Screen treatments	• Assemble graphics and text in software programming tool • Client review	• Do you build an evolutionary or throwaway prototype? • Is the user interface attractive? • Should the prototype be linear, non-linear or a mix? • Does the prototype meet the performance requirements? • Do the performance requirements need to be modified for accuracy and completeness?	• Programmer • Graphic artist • Lead HPT	• Approved proof of concept prototype

Phases	Input	Process	Decisions	Responsibility	Output
5. Detailed Prototype (narrow and deep)	• Proof of concept prototype	• Code detailed representative functionalities • Client review	• What additional functionality should the prototype illustrate in detail? • Does the prototype meet the performance requirements? • Do the performance requirements need to be modified for accuracy and completeness?	• Programmer • Lead HPT • Graphic artist	• Approved detailed prototype
6. Integrated Prototype	• Detailed Prototype	• Transfer programming from prototyping language to operational language • Client review	• Does the integrated prototype work as expected? • Does the prototype meet the performance requirements? • Do the performance requirements need to be modified for accuracy and completeness?	• Programmer • Lead HPT • Project Manager	• Approved integrated prototype

PROJECT METRIC SPREADSHEET CASE STUDY



Activities	Description	Responsibility	Inputs	Output	Estimated Proto Time	Actual Proto Time	Variance	Estimated Dev Time	Actual Dev Time	Variance
1. Create storyboard template	Design a model for conveying program content to the program with minimal ambiguity	Senior ID	Program requirements / outline	Storyboard template	2	2	0	0	0	0
2. Write storyboard	Enter content into template.	ID	Outline	Storyboard	3	8	-5	40	32	8
3. Review Storyboard	Ensure that content in storyboards is clear to programmer, instructionally effective, and does not introduce new functionalities.	Sr. ID, ID, and Programmer	Storyboards	Suggested revisions	4	6	-2	6	2	4
4. Create graphics	Develop the graphic files specified in storyboard.	Graphic artist	Storyboards	Graphic files	20	20	0	50	40	10
5. Record audio / shoot video	Develop the audio and video files specified in storyboard.	Multimedia specialist	Storyboards	Audio files	8	8	0	0	0	0
6. Format text in Word	Apply fonts, styles, formatting to prepare text for direct cutting and pasting into Director.	ID / Admin	Unformatted text	Formatted text	0	0	0	4	4	0
7. Review media	Ensure that audio, video and graphics are importable and high quality.	Programmer	Media	Usable media	2	4	-2	0	0	0
8. Digitally edit video and audio clips	Remove “um’s,” “ah’s” and unnecessary pauses to create more engaging video sequences.	Multimedia specialist	Video and audio	Edited video and audio	2	3	-1	12	9	3

Activities	Description	Responsibility	Inputs	Output	Estimated Proto Time	Actual Proto Time	Variance	Estimated Dev Time	Actual Dev Time	Variance
9. Import and align graphics & media	Bring audio, video and graphic files into Director cast and assenble media on stage.	Programmer	Media files	Cast members	12	16	-4	70	50	20
10. Find timing info. for audio & video clips	Locate the times in each video and audio clip the corresponding text boxes should appear.	Programmer / ID	"soundTime" and "MovieTime" buttons in score	Synched quote boxes	16	16	0	40	34	6
11. Cut & paste text in Director	Bring Word text objects into Director cast.	ID	Formatted Word text	Director text	0	0	0	4	1	3
12. Synch text, graphics, audio & video to voiceovers (final prototype assembly)	Time the text boxes to appear at the appropriate time as the audio / video are playing.	Programmer	Voiceovers	Finalized section of program	8	8	0	40	32	8
13. Programmer Test	Ensure that the program works from start to finish.	Programmer	Program	Tested Program	2	2	0	12	8	4
14. ID Test	Conduct a detailed review to ensure that items appear at the appropriate time, and that all navigation works as intended.	ID	Program	Tested Program	2	2	0	12	8	4

Summary	Estimated Proto Time	Actual Proto Time	Variance	Estimated Dev Time	Actual Dev Time	Variance
Average time to create one lesson (in hours)	79 hrs.	93 hrs.	+14 hrs.	30.9 hrs.	23.6 hrs.	- 7.3 hrs.

RESEARCH BASE AND BIBLIOGRAPHY

Abelow, D. (1993, March). Wake up! You've entered the transition zone. Computer Language, 10 (3), 40.

Andresino, M. (1995). *What's It Gonna Cost? Putting a Price Tag on Performance Support.* Paper presented at the Performance Support Conference, Washington, DC.

Appleton, E. (1993, July 15). Put usability to the test. Datamation, 39, (14), p. 61.

August, J. H. (1991). Joint application design: The group session approach to system design. Englewood Cliffs, NJ: Yourdon Press.

Boettcher, J. (1999). *How Much Does It Cost to Develop a Distance Learning Course? It All Depends....* Web-published column, http://www.cren.net/~jboettch/#wri

Brandon-hall.com, (2002, March) E-Learning Simulations: Tools and Services for Creating Software, Business, and Technical Skills Simulations.

Bugg, P. W. & Crejci, B. (1993, April). Object-oriented design for computer-based instruction. Paper presented at annual meeting of the International Society for Performance Improvement. Chicago, IL.

Capozzi, M. (2000). e-Learning That Starts with the Learner, Not the "e." TechTrends, 44(5), 37-39.

Carmel, E.; Whitaker, R. D. & George, J. F. (1993). PD and joint application design: A transatlantic comparison. Communications of the ACM, 36 (4), 40-49.

Comaford, C. (1994, February 7). Timeboxing lets you whistle while you work. PC Week, 11 (5), 58.

Computerworld. (1999), Software is too hard to use; A group of users and vendors is seeking, for the first time, to standardize ease-of-use ratings. Computerworld August 23, 1999 p64(1)

Foshay, R. (1994). *The Problem with ISD Models*. Paper presented at the Annual ISPI Conference, San Francisco, CA.

Gale, S. (2001) Use it or lose it (testing the usability of web-based training sites) Online Learning 5(7), 34.

Getting To 50:1 (1999-2001). Education-Related News Digest Online periodical, http://www.cogsim.com/ed_related_news/ed_digest_12.htm

Glen, R. (1993, November). RAD requires radical rethink. I.T. Magazine, 25 (11), 36.

Greer, M. (1992). ID project management: Tools and techniques for instructional designers and developers. Englewood Cliffs, NJ: Educational Technology Publications.

Greer, M. (1999). *Estimated Instructional Development Time.* Web-published article, www.michaelgreer.com/id-time.htm

Griffiths, M. (1993) Rapid Prototyping Options Shrink Development Costs. Modern Plastics, 70(9), p.45.

Hough, D. (1993). Rapid delivery: An evolutionary approach for application development. IBM Systems Journal, 32, 397-415.

Johnson, A. (1994, October 19). Usability is much more than mice and menus. PC User, (245), 49.

Jones, T. S., & Richey, R. C. (2000). Rapid prototyping methodology in action: A developmental study. Educational Technology Research and Development, 48(2), 63-80.

Karat, C. (1992, November). Cost-justifying human factors support on software development projects. Human Factors Society Bulletin, 35 (11), 1-4.

Kremers, A. (1993, August). Usability on trial. PC World, 11 (8), L4.

Liang, Y. D. D. (2001). Rapid Java application development using Jbuilder 4/5. New York: Prentice Hall.

Martin, J. (1990, March 12). Within the timebox, development deadlines really work. PC Week, 7 (10), 54.

Martin, J. (1991). Rapid application development. New York: Macmillan Publishing Company.

Merlyn, V. (1991, February). Is RAD 'RAD?' Software Magazine, 11 (2), 8-10.

Nielson, J. (1993). Usability Engineering. San Diego, CA: Academic Press.

Rathbun, G. A. (1997). Reconceiving ISD: Three perspectives on rapid prototyping as a paradigm shift. (ERIC Document Reproduction Service No. ED 409 862)Reed, S. (1993, April 19). IS managers turn to in-house usability tests: Testing ensures that custom-built applications meet business and user objectives. InfoWorld, 15 (16), 72.

Rosenberg, B. F. & Zimmerman, R. M. (1993). Accelerated application engineering: A cost-effective development approach. Information Systems Management, 10 (1), 7-14.

Schneiderman, B. (1995) Crossing disciplines: Usability as a bridge between system, software, and documentation. STC Journal, 2nd quarter Stone, D. S. (1997). Costs and Evaluation. Paper presented at the HPT Institutes. San Francisco, CA and Chicago, IL.

Teague, P. E. (1996). How to design in warp speed. Design News, 51(13), 62-3.

Tennyson, R. D. & Foshay, W. R. (2000). "Instructional Systems Development" (pp. 111- 147) in Tobias, S. and Fletcher,D. (eds.), Training & Retraining: A Handbook for Business, Industry, Government and the Military. New York: McGraw-Hill.

Tessmer, M. (1994). Formative evaluation alternatives. Performance Improvement Quarterly, 7, 3-18.

Trapp, P. R. (1991, October). Rapid prototyping as an alternative to the pilot. Modern Office Technology, 74-75.

Trevor-Deutsch, Lawry (1995). *The Business Case for New Media Learning.* Paper prepared for Industry Canada Information Technologies Industries Branch, Ottawa, ON.

Tripp, S. D. & Bichelmeyer, B. (1990). Rapid prototyping: An alternative instructional design strategy. Educational Technology Research and Development, 38, 31-44.

Tripp, S. D. (1991, February). Rapid prototyping: Methodology and methods. Paper presented at annual meeting of AECT. Orlando, FL.

Tuikka, T & Salmela, M (1998). Facilitating designer-customer communication in the World Wide Web. Internet Research: Electronic Networking Applications and Policy, 8(5), 442-451.

Villachica, S. W. & Moore, A. (1996). Using Technology to Leverage Performance. Workshop presented at the ISPI Conference, Anaheim, CA.

Villachica, S. W., & Moore, A. (1997, April). You want it when? Five proven techniques to decrease development time, improve quality, and increase user satisfaction. Paper presented at the annual conference of the International Society for Performance and Instruction. Anaheim, CA.

Villachica, S. W., & Stone, D. L. (1998). Rapid application development for performance technology: Five strategies to deliver better interventions in less time. In P. J. Dean and D. E. Ripley (Eds.), Performance improvement interventions: Performance technologies in the workplace (pp. 343-399). Washington, DC: International Society for Performance Improvement.

VNU Business Media (2001, February 27). Getting to 50:1. Online Learning News, 3 (49). Available at http://www.vnulearning.com/archive/oln022701.htm

Weston, C. et al. (1997) The Influence of Participants in Formative Evaluation on the Improvement of Learning from Written Instructional Materials. Instructional Science. 25(5) p.369-86.

Zemke, R. (1997). How Long Does It Take? (The Sequel). *Training, 34* (5), 69-70, 72,74, 76, 78.