The Economics of Software Process Improvement

Capers Jones, Chief Scientist Emeritus

www.SPR.com
CJones@SPR.com

November 21, 2005

PREVIEW OF SOFTWARE IMPROVEMENT GUIDELINES

DO

- Think long range: 3 to 5 years
- Consider all factors:
  - Management
  - Process
  - Tools
  - Organization
  - Skills and training
  - Programming Languages
  - Environment
- Plan expenses of up to $15,000 per staff member
- Consider your enterprise culture

DON'T

- Expect immediate results
- Concentrate only on Agile methods or any other "silver bullet"
- Expect major improvements for minor expenses
- Ignore resistance to change

SOFTWARE PROCESS IMPROVEMENT ISSUES

- **What are the benefits of a process improvement program?**
- **What are the risks of a process improvement program?**
- **How much can development productivity be improved?**
- **How much can maintenance productivity be improved?**
- **How much can schedules be improved?**
- **How much can quality be improved?**
- **How much will the improvements cost?**
- **How long will the improvements take?**
- **What is the ROI of a successful process improvement program?**

PROCESS IMPROVEMENT BENEFITS AND RISKS

Benefits:

- Better customer satisfaction
- Better staff morale
- Reduced risk of project failures

Risks:

- Spending money without achieving positive results
- Selecting methods that don’t work for all projects
- Expecting major improvements in one year

PROCESS IMPROVEMENT RANGES

- **Development productivity: 15% to 50% ***
- **Maintenance productivity: 25% to 60% ***
- **Schedule reductions: 10% to 70% ***
- **Quality of software: 25% to 90%**
- **Customer satisfaction: 15% to 30%**
- **Staff morale: 10% to 25%**

*** High levels of reuse are required for maximum benefits**

SOFTWARE REUSE

- **Reuse provides less than 20% of software functions in 2005.**
- **As of 2005 reuse problems sometimes outweigh value.**
- **Some reuse problems are Federal: taxes on reused materials.**
- **By 2015 reuse value should outweigh problems.**
- **Major sources of reusable components circa 2005:**

  Purchased Packages
  Legacy In-house applications
  Custom applications
  Commercial reusable components
  Objects
  Mergers and acquisitions

MAJOR COMPONENT REUSE PROBLEMS

- **Lack of standard interfaces**
- **Random update schedules by vendors**
- **Marginal quality of software components and products**
- **Regression of past functions**
- **Steep learning curve of object-oriented techniques**
- **Lack of reusable test plans and test suites**
- **Potential tax liabilities under IRS rule 401**

HOW IMPORTANT IS SOFTWARE PROJECT MANAGEMENT?

ACTIVITY	SUCCESSFUL PROJECTS	CANCELED PROJECTS
Sizing	Good	Poor
Planning	Very Good	Very Poor
Estimating	Very Good	Very Poor
Milestone tracking	Good	Very Poor
Measurement	Good	Very Poor
Change control	Excellent	Poor
Quality	Excellent	Very Poor
Risk Analysis	Good	Very Poor
Overall	Very Good	Very Poor

TECHNICAL REASONS FOR SOFTWARE FAILURES

Unsuccessful Projects	Successful Projects
No automated sizing tools	Automated sizing tools
No automated estimation tools	Automated estimation tools
No automated planning tools	Automated planning tools
No progress reporting	Accurate progress reporting
Inaccurate cost collection	Accurate cost collection
No measurement data	Substantial measurement data
Inaccurate metrics	Accurate metrics
No design reviews	Formal design reviews
No code inspections	Formal code inspections
No defect tracking	Formal defect tracking
Informal change control	Formal change control
Unstable requirements (>30%)	Stable requirements (< 10%)

SOCIAL REASONS FOR SOFTWARE FAILURES

Unsuccessful Projects

- Excessive schedule pressure
- Severe friction with clients
- Poor communications
- Divisive politics
- Naive senior executives
- Management malpractice
- Technical malpractice
- Untrained Generalists

Successful Projects

- Realistic schedule expectation
- Cooperation with clients
- Good communications
- Politics held in check
- Experienced senior executives
- Capable management
- Capable technical staff
- Trained Specialists

- Quality Assurance
- Testing
- Planning and Estimating

OTHER CORRELATIONS WITH SOFTWARE FAILURES

Intermittent Failure Factors

Geographic separation of team with inadequate communication

Multiple sub-contractors involved with inadequate communication

Extraordinary storage or timing constraints

Projects using “low bid” as sole contract criterion

Staffing build up > 15% per month

Staff attrition > 40% of project team

Abrupt introduction of new technologies

Projects by companies that are downsizing

IMPROVING SOFTWARE PRODUCTIVITY AND QUALITY

- **Start with an assessment and baseline to find out what is right and wrong with current practices.**
- **Commission a benchmark study to compare your performance with best practices in your industry**
- **Stop doing what is wrong.**
- **Do more of what is right.**
- **Set targets: *Best in Class*, Better than Average, Better than Today.**
- **Develop a three-year technology plan.**
- **Include: capital equipment, offices, tools, methods, education, culture, languages and return on investment (ROI).**

QUANTITATIVE AND QUALITATIVE GOALS

What It Means to be *Best In Class*

1. **Software project cancellation due to major overruns = zero**
2. **Software cost overruns < 5% compared to formal budgets**
3. **Software schedule overruns < 3% compared to formal plans**
4. **Development productivity > 25 function points per staff month**
5. **Software reuse of design, code and test cases averages > 75%**
6. **Development cost < $500 per function point at delivery**
7. **Development schedules average 25% shorter than industry average**

QUANTITATIVE AND QUALITATIVE GOALS (cont.)

8. **Software defect potentials average < 2.5 per function point**
9. **Software defect removal efficiency averages > 96% for all projects**
10. **Software delivered defects average < 0.1 per function point**
11. **Software maintenance assignment scopes > 3,500 function points**
12. **Annual software maintenance < $75 per function point**
13. **Customer service: Best of any similar corporation**
14. **User satisfaction: Highest of any similar corporation**
15. **Staff morale: Highest of any similar corporation**
16. **Compensation and benefits: Best in your industry**

U.S. SOFTWARE PERFORMANCE LEVELS

PROJECT MANAGEMENT		TECHNICAL STAFFS		SOFTWARE USERS	
Sizing	Fair	Requirements	Fair	Requirements	Poor
Estimating	Poor	Design	Good	Schedule Demands	Poor
Planning	Fair	Coding	Good	Reviews	Fair
Tracking	Poor	Reviews	Fair	Acceptance Test	Fair
Measuring	Poor	Testing	Good	Usage	Good
Overall	Poor		Good		Fair

Conclusion: U. S. technical skills are better than U. S. management skills.
Project management and quality are frequent problem areas.

RESULTS ON >10,000 FUNCTION POINT PROJECTS

Worst-case Scenario

	Probability of Selected Outcomes			
	Cancel	**Delays**	**On time**	**Early**
1) Manual estimates Manual plans Informal tracking Minimal quality control	40%	45%	15%	0%

Litigation probability > 50%
if project is done under contract.

RESULTS ON >10,000 FUNCTION POINT PROJECTS

Single-factor Scenarios

	Probability of Selected Outcomes			
	Cancel	**Delays**	**On time**	**Early**
2) Manual estimates **Automated plans** Informal tracking Minimal quality control	37%	42%	20%	1%
3) Manual estimates Manual plans **Formal tracking** Minimal quality control	35%	39%	24%	2%
4) **Automated estimates** Manual plans Informal tracking Minimal quality control	33%	36%	28%	3%
5) Manual estimates Manual plans Informal tracking **Optimal quality control**	30%	32%	34%	4%

RESULTS ON >10,000 FUNCTION POINT PROJECTS

Two-factor Scenarios

	Probability of Selected Outcomes			
	Cancel	**Delays**	**On time**	**Early**
6) Manual estimates **Automated plans** **Formal tracking** Minimal quality control	27%	28%	40%	5%
7) **Automated estimates** **Automated plans** Informal tracking Minimal quality control	23%	26%	45%	6%
8) **Automated estimates** Manual plans **Formal tracking** Minimal quality control	20%	23%	50%	7%

RESULTS ON >10,000 FUNCTION POINT PROJECTS

Two-factor Scenarios

	Probability of Selected Outcomes			
	Cancel	**Delays**	**On time**	**Early**
9) Manual estimates **Automated plans** Informal tracking **Optimal quality control**	18%	20%	54%	8%
10) Manual estimates Manual plans **Formal tracking** **Optimal quality control**	16%	17%	58%	9%
11) **Automated estimates** Manual plans Informal tracking **Optimal quality control**	13%	15%	62%	10%

RESULTS ON >10,000 FUNCTION POINT PROJECTS

Three-factor Scenarios

	Probability of Selected Outcomes			
	Cancel	**Delays**	**On time**	**Early**
12) **Automated estimates** **Automated plans** **Formal tracking** Minimal quality control	10%	12%	67%	11%
13) Manual estimates **Automated plans** **Formal tracking** **Optimal quality control**	8%	10%	69%	13%
14) **Automated estimates** Manual plans **Formal tracking** **Optimal quality control**	5%	8%	72%	15%
15) **Automated estimates** **Automated plans** Manual tracking **Optimal quality control**	3%	6%	74%	17%

RESULTS ON >10,000 FUNCTION POINT PROJECTS

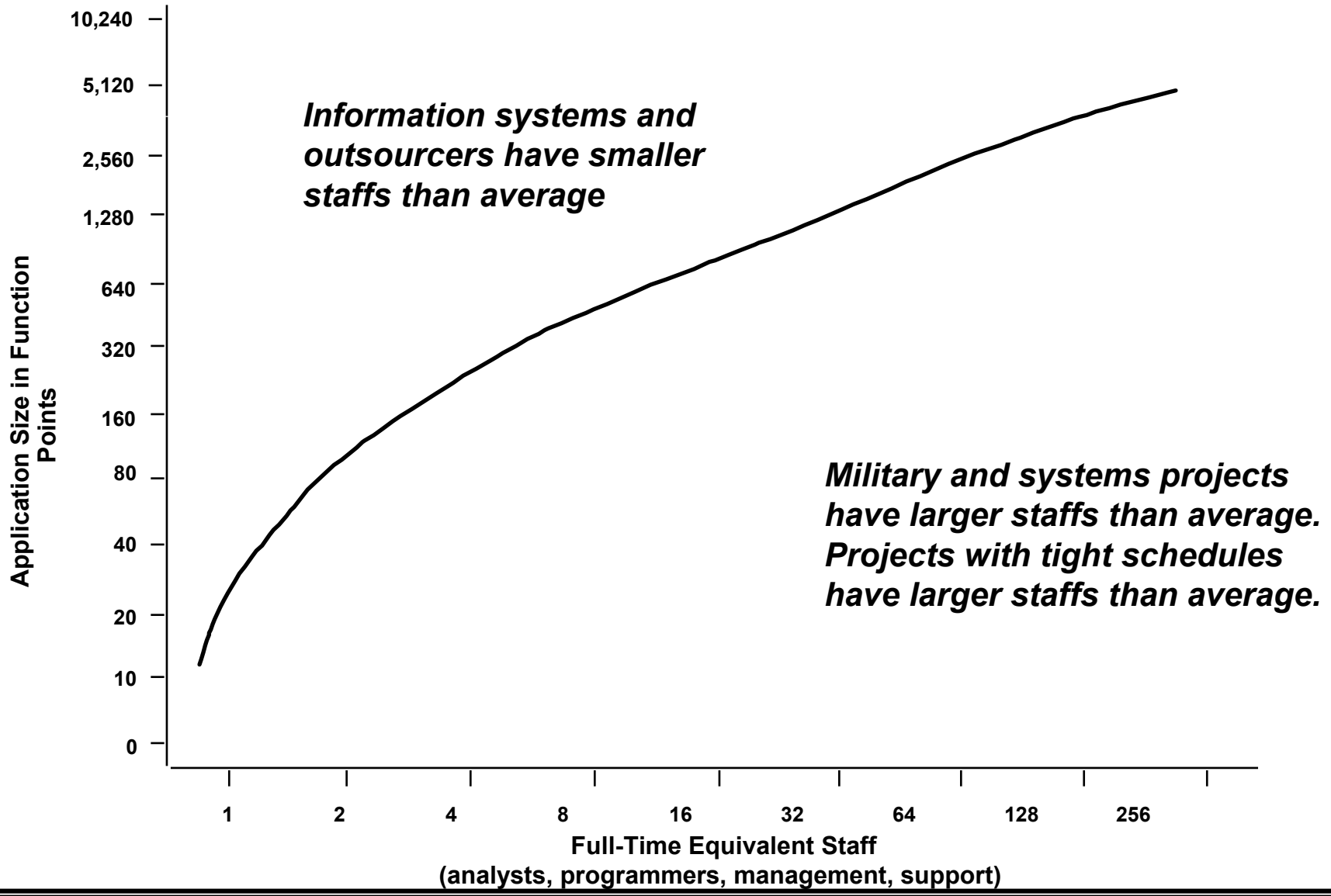
Best-case Scenario

	Probability of Selected Outcomes			
	Cancel	**Delays**	**On time**	**Early**
16) **Automated estimates** **Automated plans** **Formal tracking** **Optimal quality control**	1%	2%	78%	19%

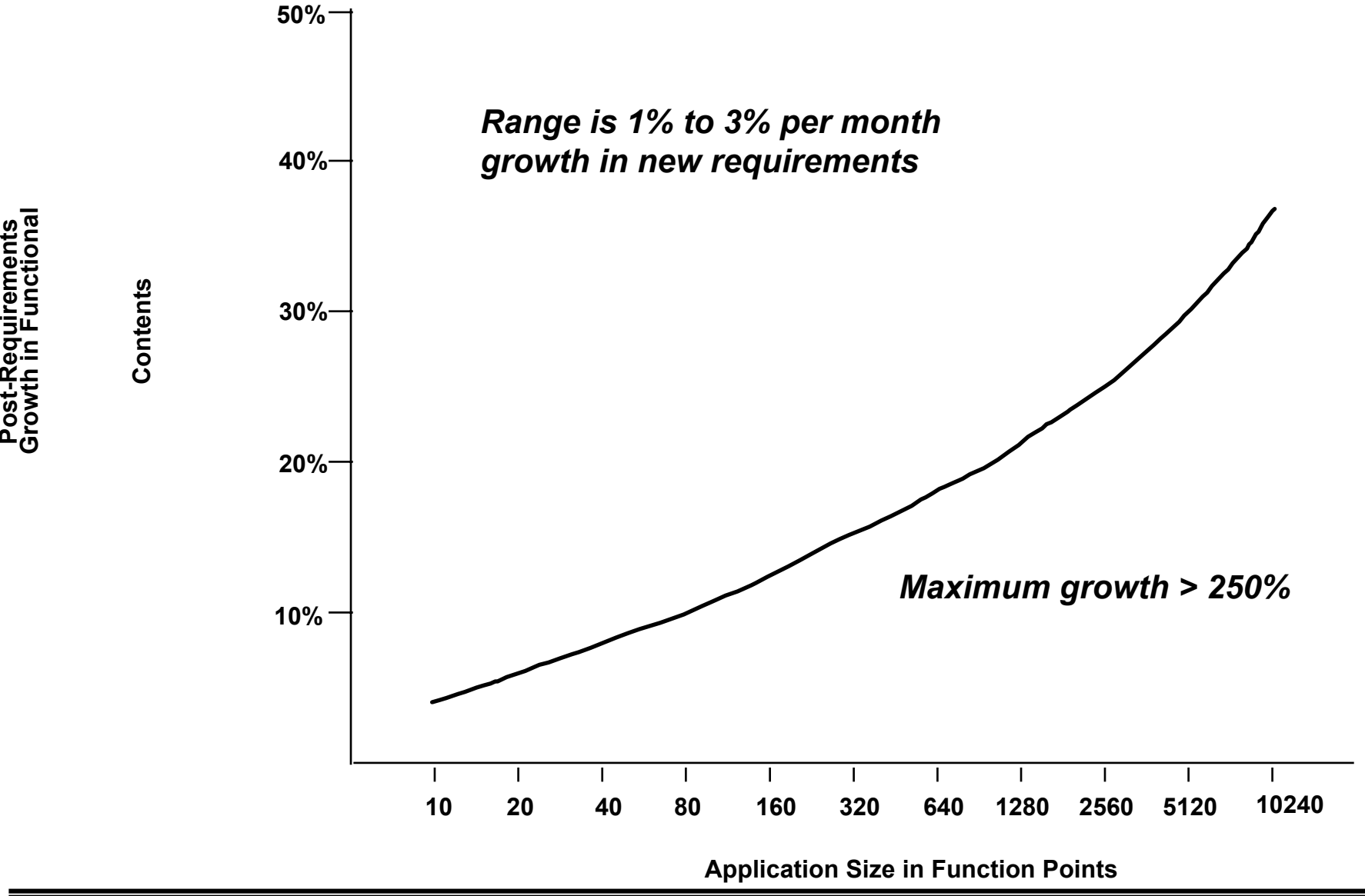
BODY OF KNOWLEDGE FOR SOFTWARE PROJECTS

- **Problem 1: Requirements grow at > 2% per month**
- **Problem 2: Defects and errors often > 5 per function point**
- **Problem 3: Cumulative defect removal often < 85%**
- **Problem 4: Testing defect removal often < 70%**
- **Problem 5: Average schedules => function points ^ 0.4 power**
- **Problem 6: Clients expect schedules of =< function points ^ 0.3**
- **Problem 7: Above 10,000 function points project failures > 50%**
- **Problem 8: Most delays due to poor quality and requirements creep**
- **Problem 9: Project Managers not equipped to defend estimates**

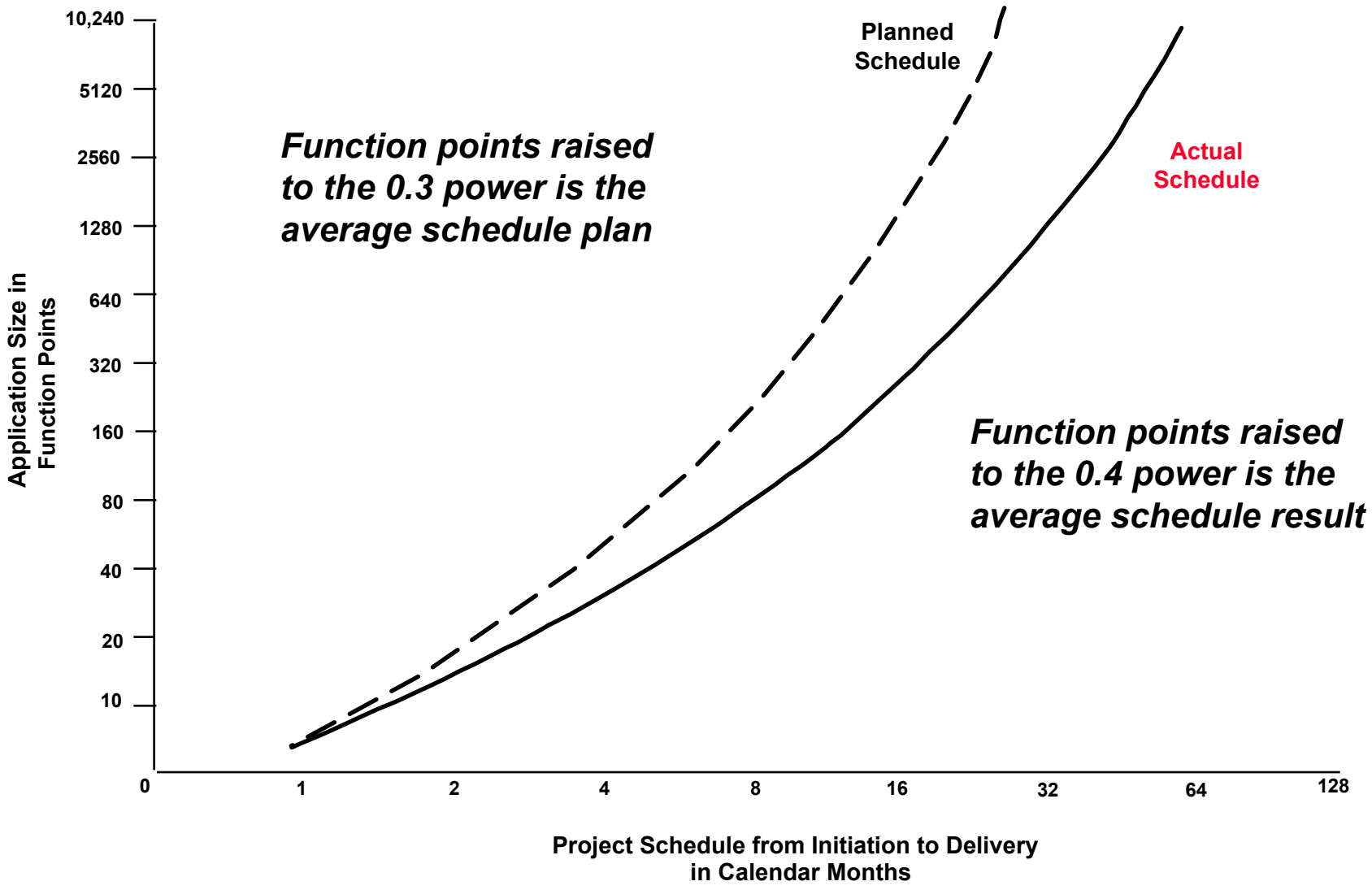
AVERAGE SOFTWARE STAFF SIZES



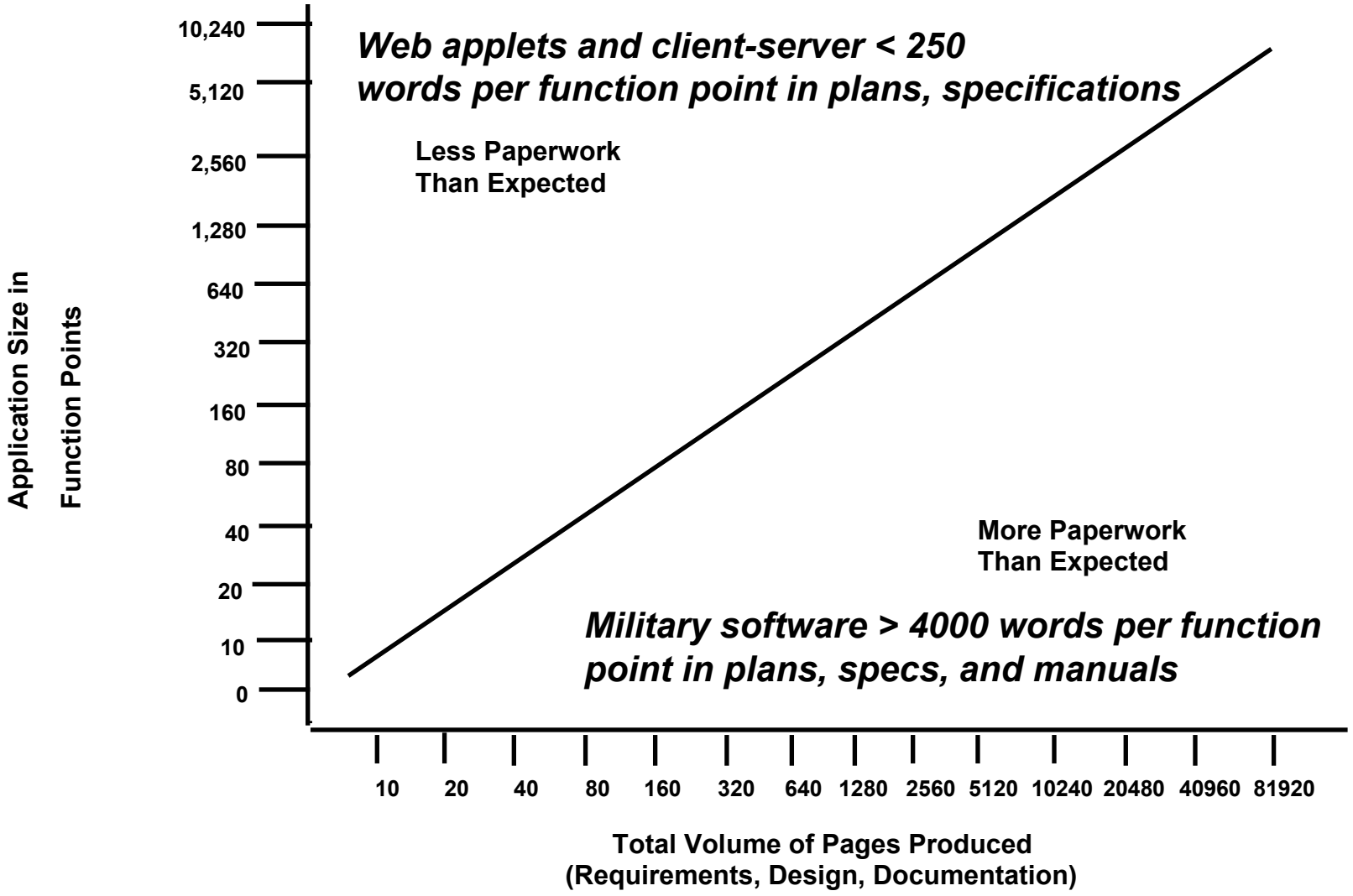
GROWTH OF NEW FUNCTIONS AFTER REQUIREMENTS



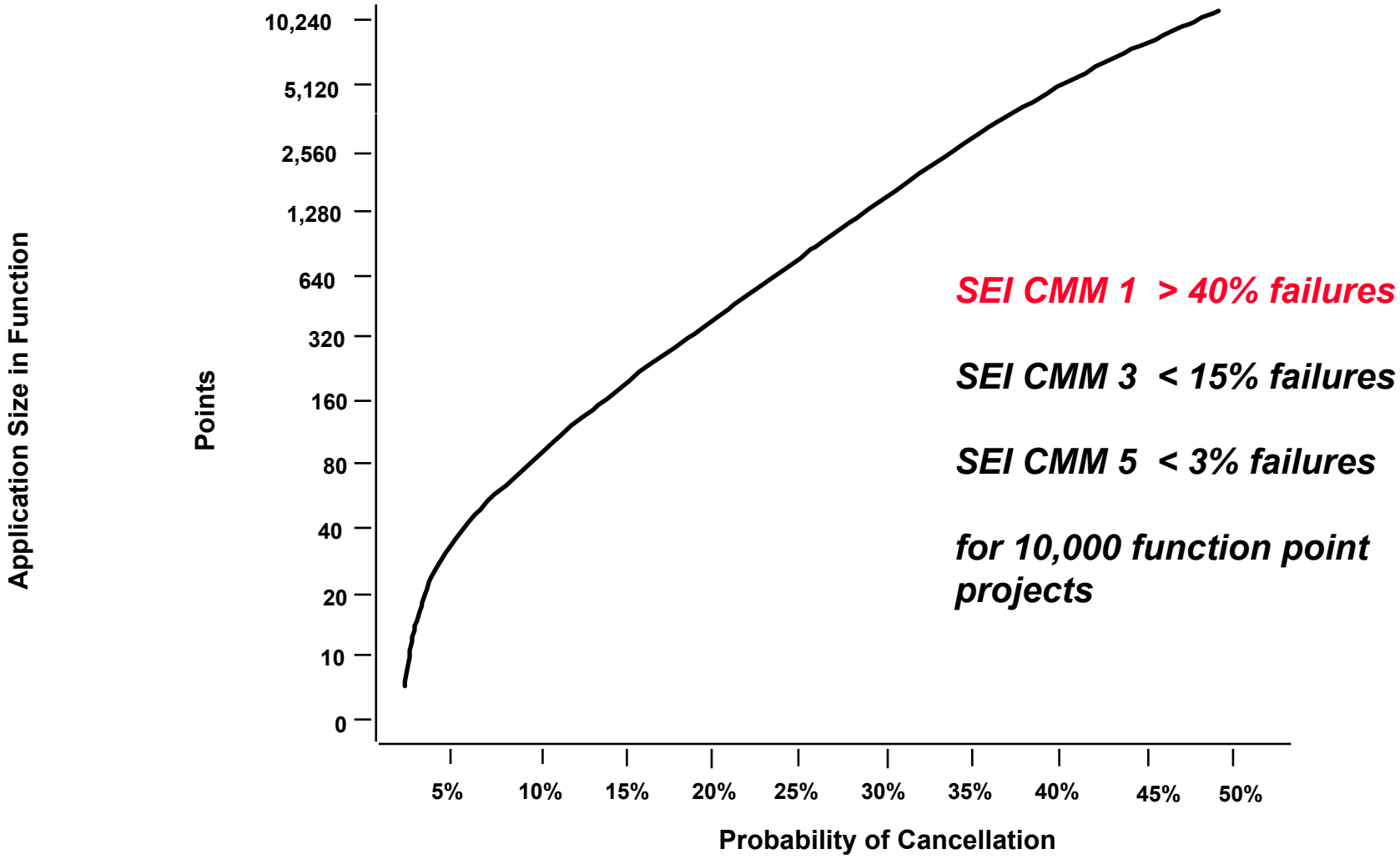
PLANNED VERSUS ACTUAL PROJECT SCHEDULES



SOFTWARE PAPERWORK



RISK OF PROJECT FAILURE

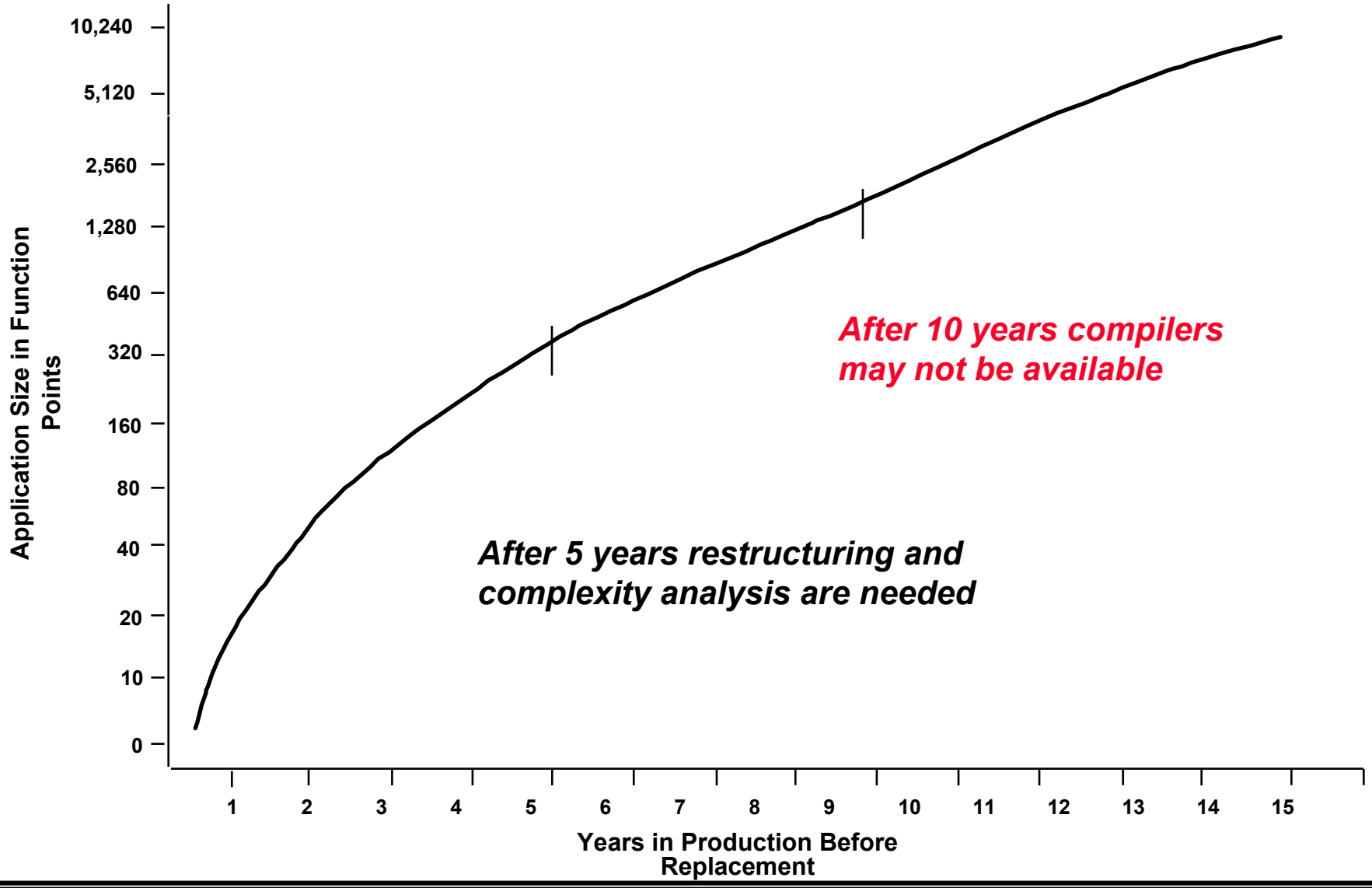


RISKS OF FAILURE OR DELAY BY CMM LEVEL

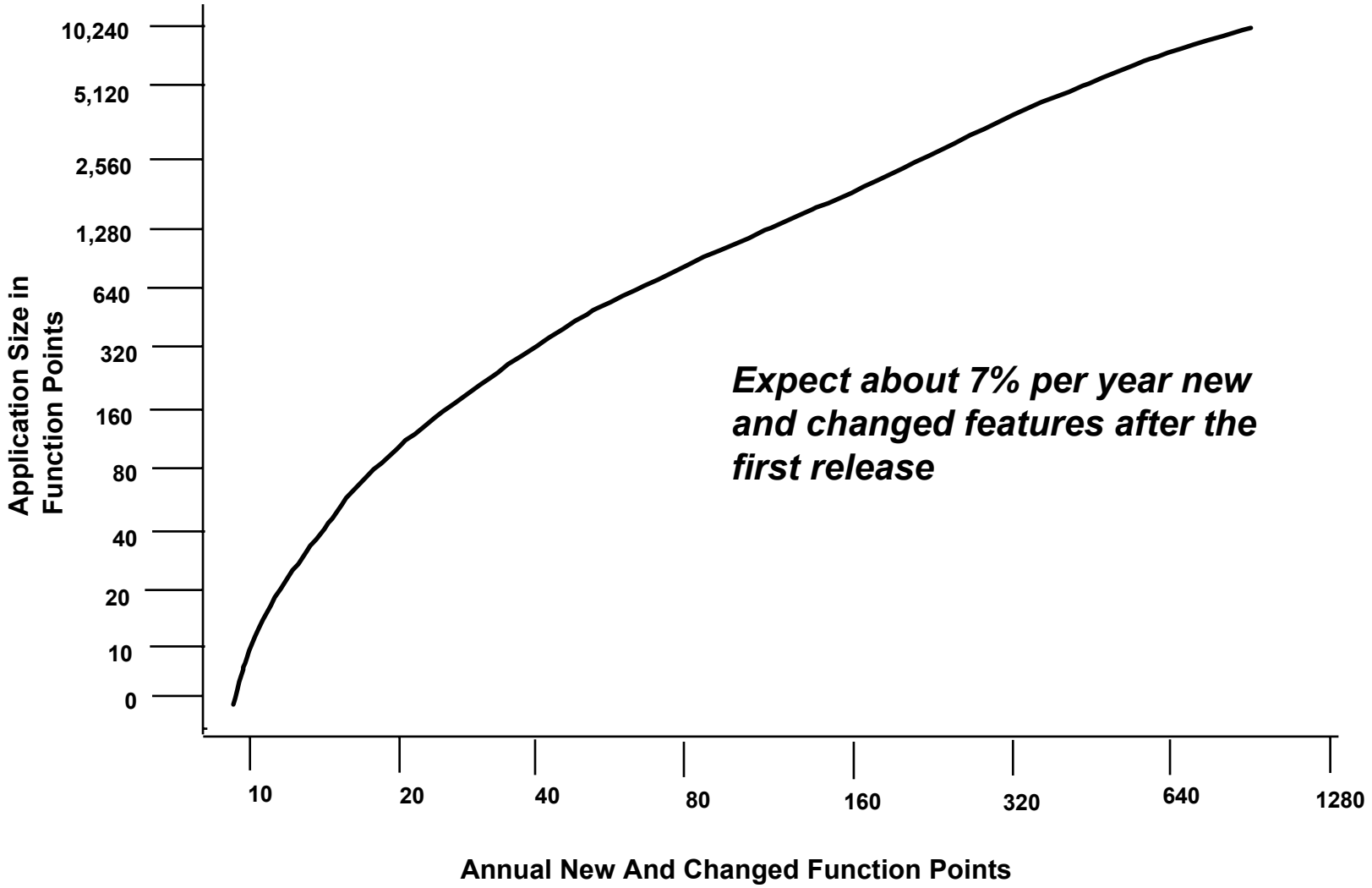
(Complex projects of 10,000 function points in size)

SEI CMM LEVEL	Delay > 1 year	Termination
SEI CMM Level 1	35%	40%
SEI CMM Level 2	30%	30%
SEI CMM Level 3	20%	15%
SEI CMM Level 4	12%	05%
SEI CMM Level 5	10%	03%

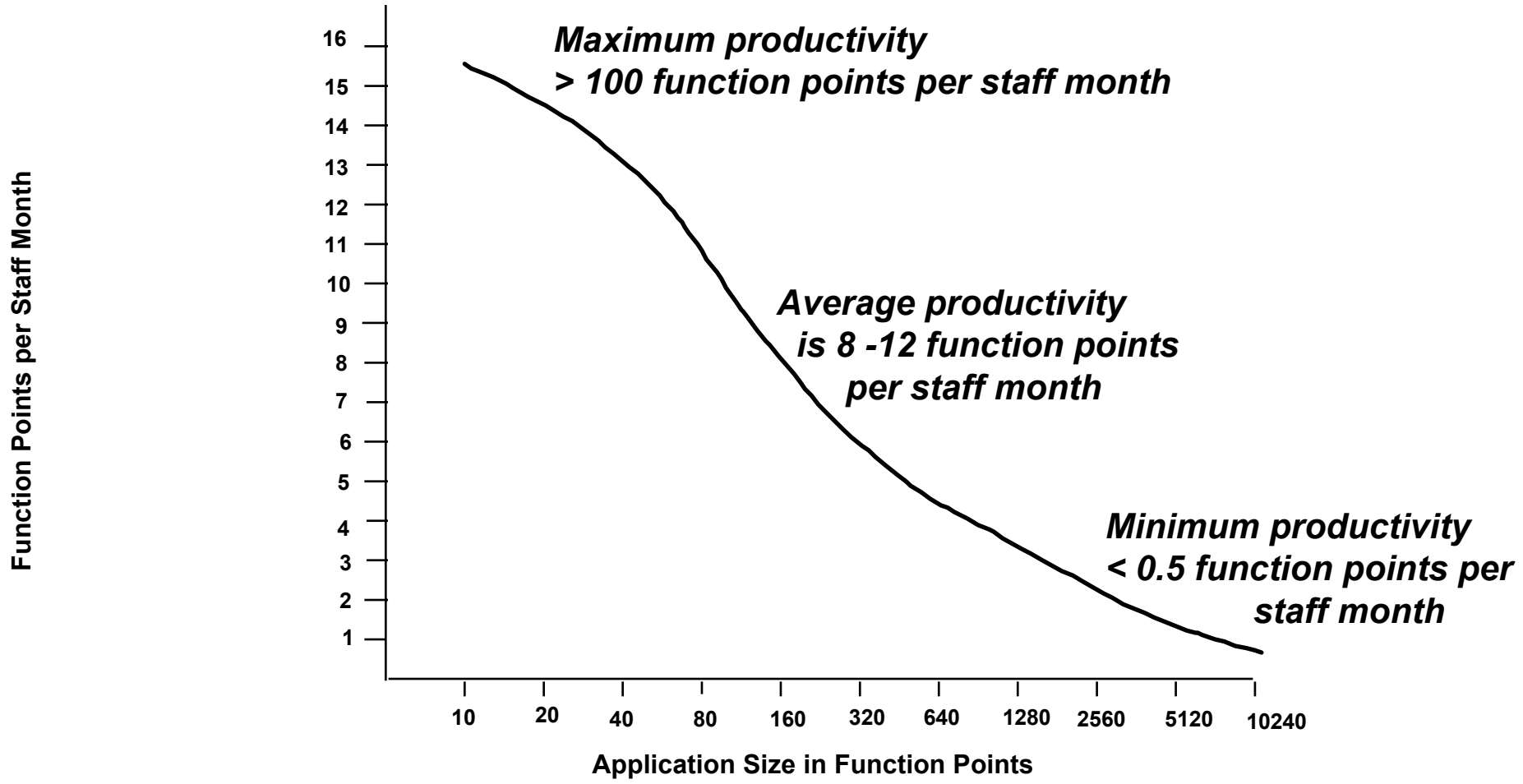
SOFTWARE LIFE EXPECTANCY



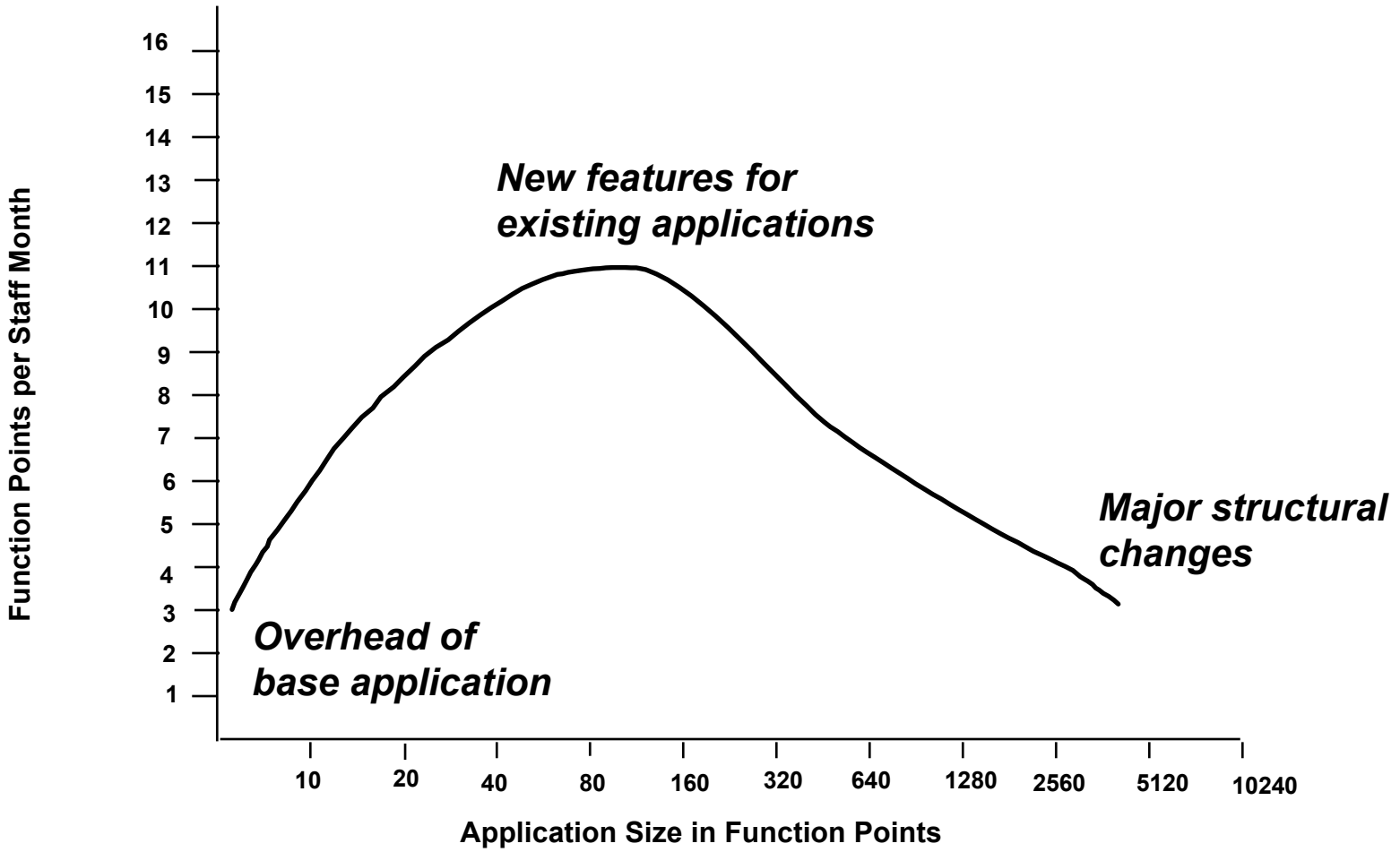
ANNUAL SOFTWARE ENHANCEMENTS



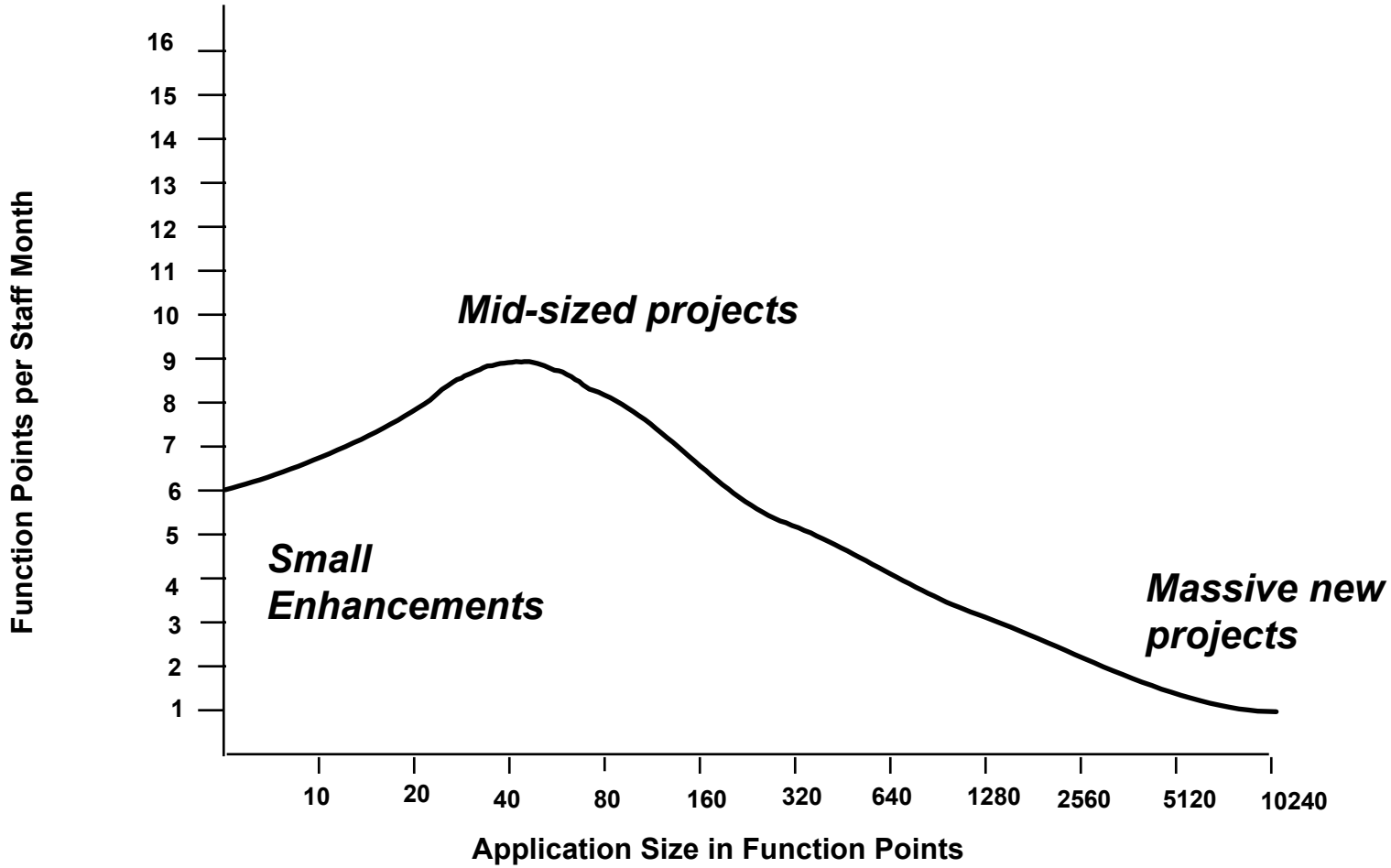
AVERAGE PRODUCTIVITY RATES (NEW PROJECTS)



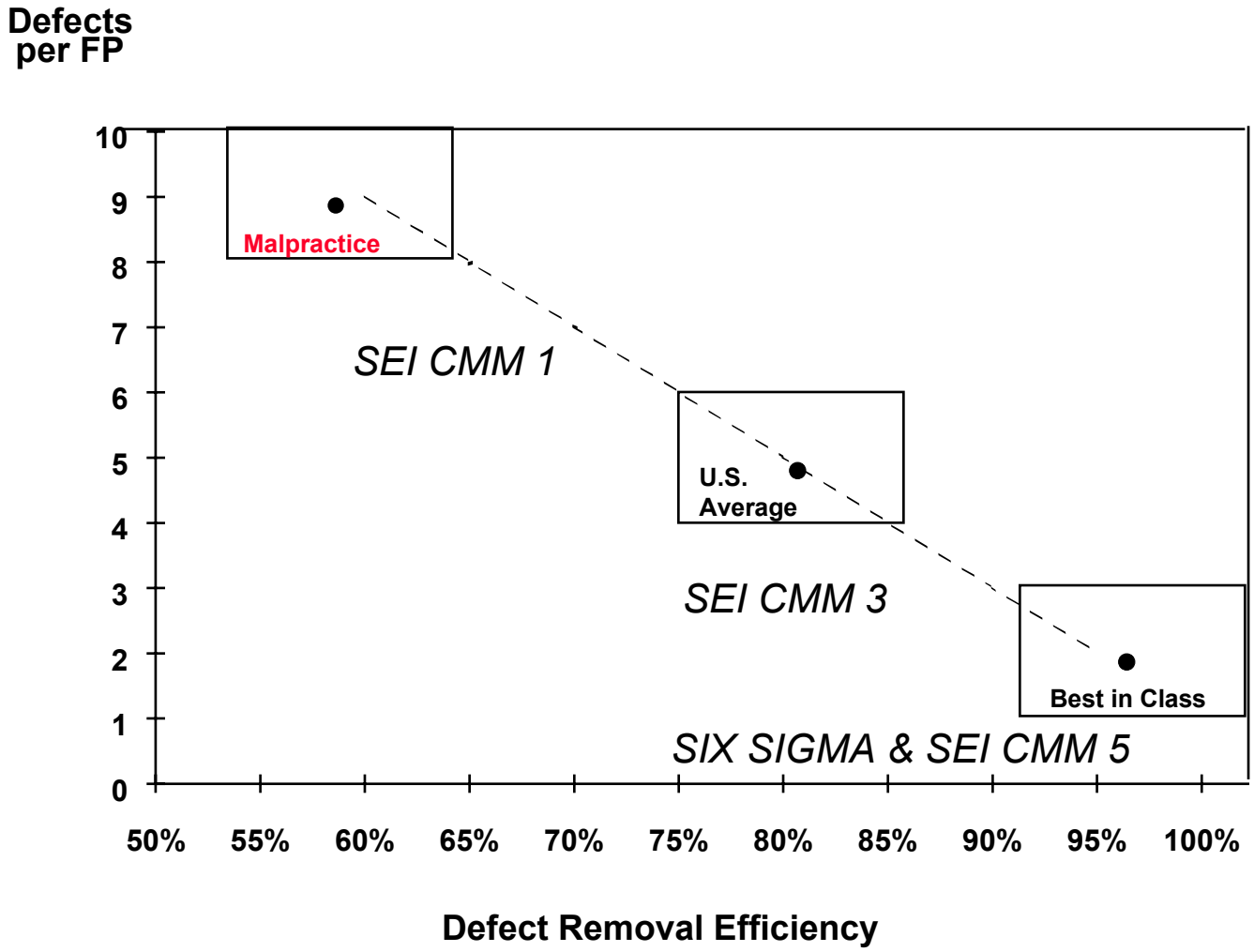
PRODUCTIVITY RATES FOR ENHANCEMENT SOFTWARE PROJECTS



PRODUCTIVITY RATES (OVERALL AVERAGE)



SOFTWARE QUALITY IMPROVEMENT



CURRENT U.S. AVERAGES FOR SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

Defect Origins	Defect Potential	Removal Efficiency	Delivered Defects
Requirements	1.00	77%	0.23
Design	1.25	85%	0.19
Coding	1.75	95%	0.09
Documents	0.60	80%	0.12
Bad Fixes	0.40	70%	0.12
TOTAL	5.00	85%	0.75

'BEST IN CLASS' RESULTS FOR SOFTWARE QUALITY

(Data expressed in terms of defects per function point)

Defect Origins	Defect Potential	Removal Efficiency	Delivered Defects
Requirements	0.40	85%	0.08
Design	0.60	97%	0.02
Coding	1.00	99%	0.01
Documents	0.40	98%	0.01
Bad Fixes	0.10	95%	0.01
TOTAL	2.50	96%	0.13

SIX STAGES ON THE PATH TO SOFTWARE EXCELLENCE

Stage 0: **Assessment, Baseline, Benchmark analysis**

Stage 1: **Focus on Project Management**

Stage 2: **Focus on Structured Methods**

Stage 3: **Focus on New Tools and Approaches**

Stage 4: **Focus on Infrastructure**

Stage 5: **Focus on Reusability**

Stage 6: **Focus on Industry Leadership**

TIME REQUIRED TO ADVANCE FROM STAGE TO STAGE

(Duration in Calendar Months)

	Enterprise Software Population			
	<10	11-100	101-1000	>1000
Stage 0 Assessment/Baseline	1	2	3	4
Stage 1 Management	3	6	7	8
Stage 2 Methods	3	6	9	9
Stage 3 Tools	4	4	6	9
Stage 4 Infrastructure	3	4	6	9
Stage 5 Reusability	4	6	8	12
Stage 6 Leadership	6	8	9	9
Total	24	36	48	60

THE QUALITY AND PRODUCTIVITY BENEFITS FROM COMPLETING EACH STAGE

	Defect Reduction	Productivity Increase	Schedule Compression
Stage 0 Assessment	0	0	0
Stage 1 Management	- 10%	0	- 10%
Stage 2 Methods	- 50%	25%	- 15%
Stage 3 Tools	- 10%	35%	- 15%
Stage 4 Infrastructure	- 5%	10%	- 5%
Stage 5 Reusability	- 85%	65%	- 50%
Stage 6 Leadership	- 5%	5%	- 5%
Overall Results	- 90%	350%	- 70%

PROCESS IMPROVEMENT EXPENSES PER CAPITA

	Small < 100 staff	Medium 100-1000	Large > 1000 staff	
Stage 0 Assessment	$125	$150	$250	*SEI CMM 1*
Stage 1 Management	$1000	$2500	$3000	
Stage 2 Methods	$1500	$2500	$3500	*SEI CMM 2*
Stage 3 Tools	$2500	$3500	$5000	
Stage 4 Infrastructure	$1500	$2000	$3000	*SEI CMM 3*
Stage 5 Reusability	$2000	$2500	$3500	*SEI CMM 4*
Stage 6 Leadership	$1000	$1000	$2000	*SEI CMM 5*
Overall Results	$9625	$14150	$20250	

PROCESS IMPROVEMENT RETURN ON INVESTMENT (ROI)

- **Assume improvement costs of about $1,000,000**
- **Value of better quality $5,000,000**
- **Value of shorter schedules $4,000,000**
- **Value of higher productivity $3,000,000**
- **Value of reduced maintenance $2,000,000**
- **Value of better customer satisfaction $1,000,000**
- **TOTAL VALUE $15,000,000 ***
- **RETURN ON INVESTMENT $15 to $1**

*** Assumes 2 years of improvements and 3 years of results**

UNSUCCESSFUL PROCESS IMPROVEMENT

- **Assume improvement costs of about $1,000,000**
- **Value of better quality $100,000**
- **Value of shorter schedules $100,000**
- **Value of higher productivity $100,000**
- **Value of reduced maintenance $100,000**
- **Value of better customer satisfaction $100,000**
- **TOTAL VALUE $500,000 ***
- **RETURN ON INVESTMENT $0.5 to $1**

*** Assumes 2 years of improvements and 3 years of results**

STAGE 0: ASSESSMENT, BASELINE, BENCHMARKS

Key Technologies

- SEI Assessment (Levels 1 through 5)
- 6 Sigma Baseline, Benchmark
- SPR Assessment, Baseline, Benchmark
- ISO 9001 - 9004 Audit
- Putnam Baseline, Benchmark
- Gartner Baseline, Benchmark
- David's Baseline, Benchmark
- IFPUG Baseline, Benchmark

STAGE 1: FOCUS ON PROJECT MANAGEMENT

Key Technologies

- Project Sizing
- Project Schedule Planning
- Project Cost Estimating
- Project Quality Estimating
- Functional Metrics
- Project Measurement
- Project Milestone Tracking
- Package Acquisition
- Risk Analysis
- Value Analysis

STAGE 2: FOCUS ON DEVELOPMENT PROCESSES

Key Technologies

- Reviews and Inspections
- Joint Application Design (JAD)
- 6 Sigma methodology
- Agile methodologies with caution
- ISO 9001 - 9004 Certification with caution
- SEI maturity levels
- Geriatric Technologies for legacy systems

STAGE 3: FOCUS ON NEW TOOLS & APPROACHES

- **Key Technologies -- New Tools**
  - **Integrated Tool suites**
  - **Multi-media Tools**
  - **Repositories**
  - **Web and Internet Tools**

- **Key Technologies -- New Approaches**
  - **Requirements analysis tools**
  - **Design tools**
  - **Object-oriented Methods**
  - **Testing tools**
  - **Reverse Engineering and maintenance tools**

STAGE 4: FOCUS ON INFRASTRUCTURE

Key Technologies

- Staff Specialization
- Formal Measurement Organization
- Formal Maintenance Organization
- Formal Quality Assurance Organization
- Formal Testing Organization
- Formal Process Improvement Organization
- Improved Hiring Practices
- Improved Compensation Plans
- Competitive Analysis
- Outsource Analysis

STAGE 5: FOCUS ON REUSABILITY

Key Technologies

- Reusable Architectures
- Reusable Requirements
- Reusable Designs
- Reusable Interfaces
- Reusable Source Code
- Reusable Plans
- Reusable Estimates
- Reusable Data
- Reusable Human Interfaces
- Reusable Test Plans
- Reusable Test Cases
- Reusable Documentation

High quality reuse has best ROI of any technology: > $40 per $1 expended.

Low quality reuse has worst ROI of any technology: > - $15 for every $1 expended.

STAGE 6: FOCUS ON INDUSTRY LEADERSHIP

Key Technologies

- **Baldrige Award**
- **Deming Prize**
- **SEI CMM Level 5 for major software sites**
- **Best 100 Companies to Work For**
- **Market share grows > 20% from baseline**
- **Time to market better than competitors by > 30%**
- **Acquisition of Competitors**
- **Become a Software Outsourcer**

ATTRIBUTES OF BEST IN CLASS COMPANIES

1. **Good project management**
2. **Good technical staffs**
3. **Good support staffs**
4. **Good measurements**
5. **Good organization structures**
6. **Good methodologies**
7. **Good tool suites**
8. **Good environments**

GOOD PROJECT MANAGEMENT

- **Without good project management the rest is unachievable**
- **Attributes of project good management:**
  - **Fairness to staff**
  - **Desire to be excellent**
  - **Strong customer orientation**
  - **Strong people orientation**
  - **Strong technology orientation**
  - **Understands planning and estimating tools**
  - **Can defend accurate estimates to clients and executives**
  - **Can justify investments in tools and processes**

GOOD SOFTWARE ENGINEERING TECHNICAL STAFFS

- Without good engineering technical staffs tools are not effective
- Attributes of good technical staffs:
  - Desire to be excellent
  - Good knowledge of applications
  - Good knowledge of development processes
  - Good knowledge of quality and defect removal methods
  - Good knowledge of maintenance methods
  - Good knowledge of programming languages
  - Good knowledge of software engineering tools
  - Like to stay at the leading edge of software engineering

GOOD SUPPORT STAFFS

- **Without good support technical staffs and managers are handicapped**
- **Support staffs > 30% of software personnel in leading companies**
- **Attributes of good support staffs:**
  - **Planning and estimating skills**
  - **Measurement and metric skills**
  - **Writing/communication skills**
  - **Quality assurance skills**
  - **Data base skills**
  - **Network, internet, and web skills**
  - **Graphics and web-design skills**
  - **Testing and integration skills**
  - **Configuration control and change management skills**

GOOD SOFTWARE MEASUREMENTS

- **Without good measurements progress is unlikely**
- **Attributes of good measurements:**
  - **Function point analysis of entire portfolio**
  - **Annual function point benchmarks**
  - **Life-cycle quality measures**
  - **User satisfaction measures**
  - **Development and maintenance productivity measures**
  - **Soft factor assessment measures**
  - **Hard factor measures of costs, staffing, effort, schedules**
  - **Measurements used as management tools**

GOOD ORGANIZATION STRUCTURES

- Without good organization structures progress is unlikely
- Attributes of good organization structures:
  - Balance of line and staff functions
  - Balance of centralized and decentralized functions
  - Organizations are planned
  - Organizations are dynamic
  - Effective use of specialists for key functions
  - Able to integrate “virtual teams” at remote locations
  - Able to integrate telecommuting

GOOD PROCESSES AND METHODOLOGIES

- **Without good processes and methodologies tools are ineffective**
- **Attributes of good methodologies:**
  - **Flexible and useful for both new projects and updates**
  - **Scalable from small projects up to major systems**
  - **Versatile and able to handle multiple kinds of software**
  - **Efficient and cost effective**
  - **Evolutionary and able to handle new kinds of projects**
  - **Unobtrusive and not viewed as bureaucratic**

GOOD TOOL SUITES

- **Without good tool suites, management and staffs are handicapped**
- **Attributes of good tool suites:**
  - **Both project management and technical tools**
  - **Functionally complete**
  - **Mutually compatible**
  - **Easy to learn**
  - **Easy to use**
  - **Tolerant of user errors**
  - **Secure**

GOOD ENVIRONMENTS AND ERGONOMICS

- **Without good office environments productivity is difficult**
- **Attributes of good environments and ergonomics:**
  - **Private office space for knowledge workers (> 90 square feet; > 6 square meters)**
  - **Avoid small or crowded cubicles with 3 or more staff**
  - **Adequate conference and classroom facilities**
  - **Excellent internet and intranet communications**
  - **Excellent communication with users and clients**

SOFTWARE IMPROVEMENT GUIDELINES

DO

- Think long range: 3 to 5 years
- Consider all factors:
  - Management
  - Process
  - Tools
  - Organization
  - Skills and training
  - Programming Languages
  - Environment
- Plan expenses of up to $15,000 per staff member
- Consider your corporate culture

DON'T

- Expect immediate results
- Concentrate only on Agile methods or any other "silver bullet"
- Expect major improvements for minor expenses
- Ignore resistance to change