

Subject: Software Engineering
Topic: Software maintenance



LISNA THOMAS

Academic year:2020-21

Software Maintenance

- ◆ Team Presentation.
- ◆ Why Maintenance?
- ◆ Problems of Maintenance.
- ◆ Design a Software to ease Maintenance.
- ◆ Conclusion.

Team Members

- ◆ *El Yazid ALAOUI YAZIDI* : Junior student in Software Engineering.
- ◆ *Abdelouhab TAOUFIQ* : Senior student in Software Engineering.
- ◆ *Youssef ALAOUI MDEGHRI* : Senior student in Software Engineering.

Why Maintenance?

- ◆ Definition of Maintenance,
- ◆ Software Maintenance Objectives,
- ◆ Enterprise Resource Planning .

Definition of Maintenance

- ◆ Is the set of activities, both technical and managerial, that ensures that software continues to meet organizational and business objectives in a cost effective way.

Software Maintenance Objectives

- ◆ Difference between Software Product and Software Maintenance is:
 - ◆ Software Product is the result of the Software development.
 - ◆ Software Maintenance results in a service being delivered to the customer.

Software Maintenance Objectives

- ◆ Corrective,
- ◆ Adaptive,
- ◆ Perfective,
- ◆ Inspection.

Types of Maintenance

◆ Corrective:

- ◆ Taking existing code and correcting a fault that causes the code to behave in some way that deviates from its documented requirements.

Types of Maintenance

◆ Adaptive:

- ◆ Taking existing code and adapting it to provide new features and functionality. These are typically part of a new release of the code and part of a larger development effort.

Types of Maintenance

◆ Perfective:

- ◆ These are typically made to improve the maintainability of the code such as restructuring it to make it more easily understood or to remove ambiguities.

Types of Maintenance

◆ Inspection:

- ◆ These are usually made as a result of code inspections and focus more of adhering to coding standards or to reduce the likelihood of a failure.

Enterprise Resource Planning

- ◆ ERP is a perfective strategy planning for most software application domains which look forward to the point where their software can be developed from existing architectures, frameworks, patterns and components.

Resource Planning objectives

- ◆ Solve future business application problems,
- ◆ Integrate those solutions across the enterprise.

Problems of Maintenance

- ◆ Maintenance Attributes,
- ◆ Maintenance Organization,
- ◆ Respect of Metrics,
- ◆ Requirements volatility.

Maintenance Attributes

- ◆ five factors drive the difficulty of delivering software:
 - ◆ Product ,
 - ◆ Documentation,
 - ◆ End users,
 - ◆ Process,
 - ◆ Environment.

Maintenance Attributes

- ◆ Important attributes to be considered during Maintenance:
 - ◆ product age,
 - ◆ design,
 - ◆ Language,
 - ◆ current failure rate,
 - ◆ staff experience.

Maintenance Organization

- ◆ five factors drive the difficulty of delivering software:
 - ◆ Input Vs. output,
 - ◆ cycle-time,
 - ◆ cost/change,
 - ◆ schedule,
 - ◆ flexibility,
 - ◆ Quality.

Respect of Metrics

- ◆ Software maintenance should be measured and managed using metrics to reach a quality software.
- ◆ However, we don't know how to measure maintainability because it's a service.
- ◆ Approaches were made to get values that can be useful during maintenance (surveys).

Survey

	Ex 1	Ex 2	Ex 3
Throughput (Changes Delivered)	209	846	139
Priority Change Response Time (Days)	111	92	108
Economics (K\$/Change)	78	116	45
% Releases w/Content Changes (total releases)	33	75	65
% Schedules Met	100	70	24
Customer-Reported Defects per Change Delivered	51	21	4

Goal	Question	Metric(s)
Maximize Customer Satisfaction	How many problems affect the customer?	Current Change Backlog Software Reliability
	How long does it take to fix an Emergency or Urgent problem?	Change Cycle Time from Date Approved and from Date Written
Minimize Cost	How much does a software maintenance delivery cost?	\$/delivery
	How are the costs allocated?	\$/activity
	What kinds of changes are being	Number of Changes by Type
	How much effort is expended per change type?	Staff Days Expended per change by type
	How many invalid change requests are evaluated?	% Invalid Change Requests Closed each quarter
Minimize Schedule	How difficult is the delivery?	Complexity Assessment Computer Resource Utilization
	How many changes are made to the planned delivery content?	% Content Changes by Delivery
	Are we meeting our delivery	% On-time deliveries

Requirements volatility

- ◆ Requirements are the foundation of the software release process.
 - ◆ Changing requirements during the software maintenance process impacts the cost, schedule, and quality of the resulting product.
 - ◆ Build model to make planning of customer communications (predictions).
- ◆ A focus is made on non volatile requirements.

Design for Maintenance

- ◆ A strategy to set a certain rules during the Software development.
- ◆ It eases the maintainability of the system.

Design Attributes

- ◆ Three main Factors that we have to ensure during the design of the Software:
 - ◆ Understandability,
 - ◆ Modifiability,
 - ◆ Stability.

Design Strategies

- ◆ Reasons to do a good Design in order to ease maintainability:
 - ◆ Low Cost for future maintenance,
 - ◆ Avoid long time periods of maintenance,
 - ◆ Ability to discover rapidly the failures.

Design Data Collection

- ◆ Collecting some characteristics during the commencement of the project about :
 - ◆ The Behavior of the requirements,
 - ◆ Case tool and the Programming Language used,
 - ◆ The environment.

Design for Maintenance

◆ Emergency Rework on Modules

- ◆ Design independent modules in order to substitute them in failures and correct the deficiencies rapidly.
- ◆ Design a structure with independent components in order to be easy in maintenance

Tips to Be Memorized

- ◆ Through many recommendations, we've noticed seven objections to action that are repeatedly raised:
 - ◆ Challenges to definitions & terms,
 - ◆ Different conclusions or effects,

Tips to Be Memorized Ctd'

- ◆ Interference,
- ◆ Different cause,
- ◆ Irrelevant reasons,
- ◆ Factor ignored,
- ◆ Counterexample.

Conclusion

- ◆ It's very hard to maintain a system than to design it.
- ◆ It's even harder to design a maintainable system, because it's difficult to predict future changes in the business environment.
- ◆ Software maintenance is a critical activity in the life cycle of a system.

References

- ◆ <http://www.kluweronline.com/issuetoc.htm/1022-7091+9+1/4+2000>
- ◆ <http://www.cs.umd.edu/~sharip/wets/papers/>
- ◆ <http://wwwsel.iit.nrc.ca/seldocs/eassedocs/WESS96Lethbridge.pdf>
- ◆ <http://wwwsel.iit.nrc.ca/seldocs/eassedocs/WESS96Singer.pdf>
- ◆ <http://wwwsel.iit.nrc.ca/seldocs/eassedocs/WESS97Lethbridge.pdf>
- ◆ <http://www.dacs.dtic.mil/topics/reliability/annalse2.pdf>
- ◆ <http://www.cs.umd.edu/~sharip/wess/papers/smith.html>