MAINTENANCE AND REENGINEERING

Software Engineering

Dr. Raj Singh

SOFTWARE MAINTENANCE

- Software is released to the end-users.
- Within days after release, issues are reported.
- Within weeks, users request enhancements, something is missing, or not working as expected.
- Within months, another user group sees benefits of software and request access or changes to fit their needs.
- Software development team continuously monitors the logs and fix potential issues.

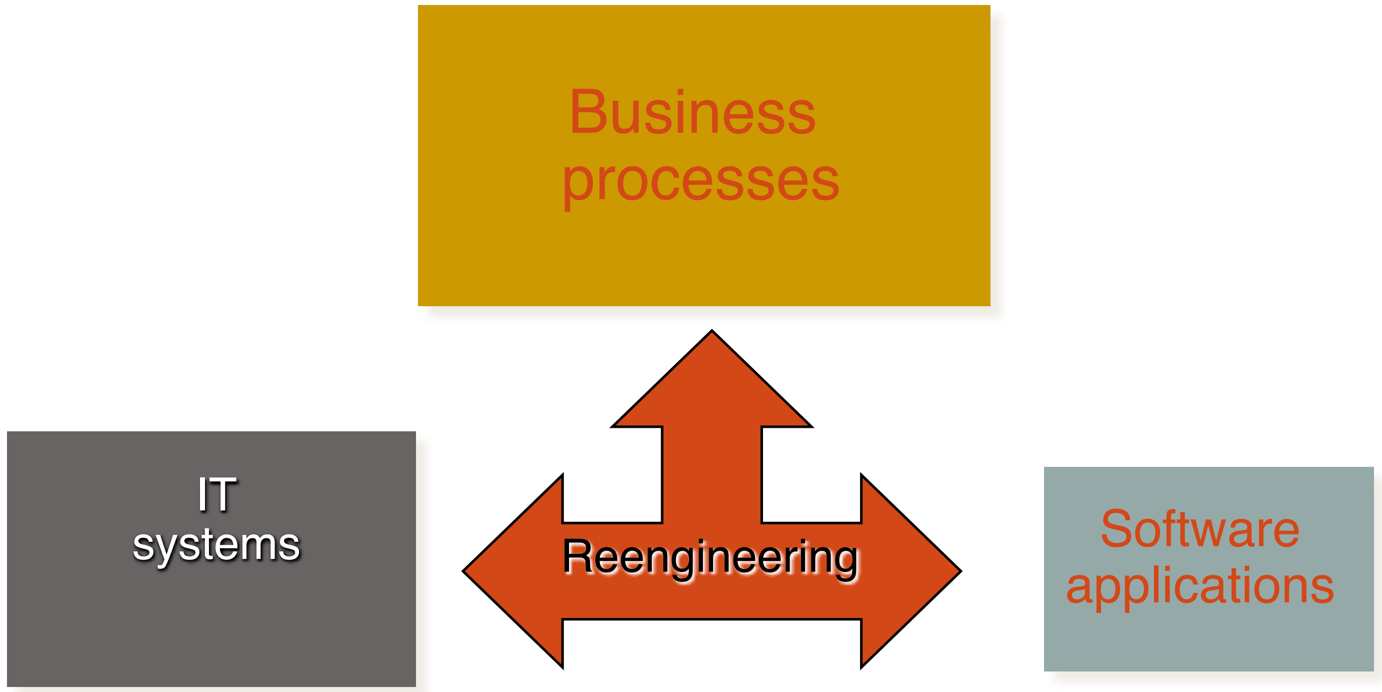
MAINTAINABLE SOFTWARE

- Maintainable software exhibits effective modularity
- It makes use of design patterns that allow ease of understanding.
- It has been constructed using well-defined coding standards and conventions leading to good quality code.
- It has undergone a variety of quality assurance techniques and problems have already been addressed.
- It has been created by software engineers who recognize that they may not be around when changes must be made.
- It is well documented and easy to make changes without breaking another applications.

SOFTWARE SUPPORTABILITY

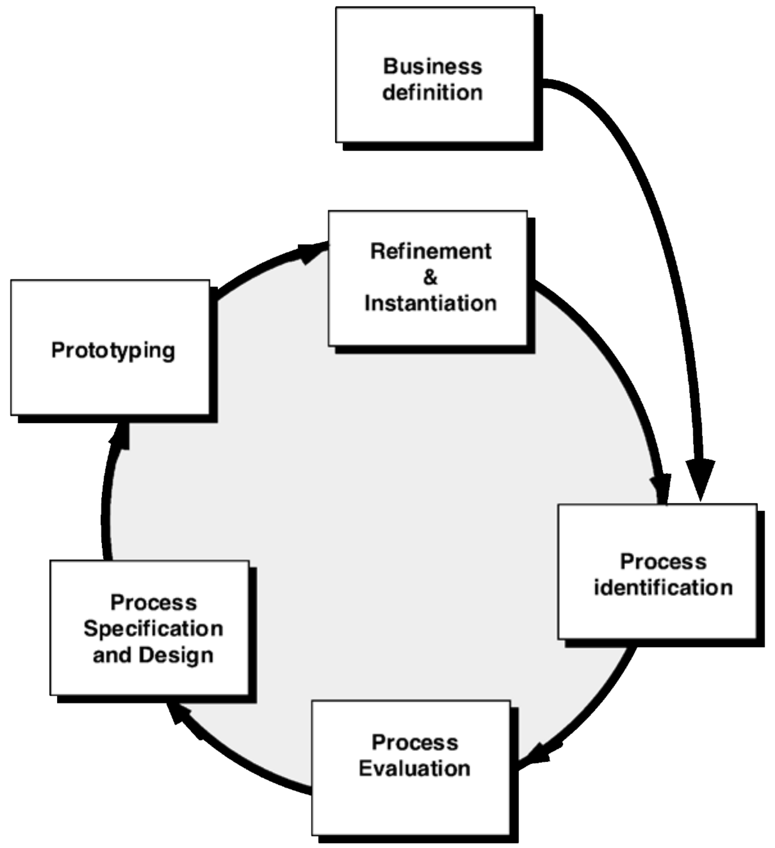
- The capability of supporting a software system over its whole product life.
- The software should contain facilities to assist support personnel when a defect is encountered.
- Support personnel have the knowledge and help documents to support the software.
- Support personnel have access to all system logs, databases, and servers to debug and fix issues.

REENGINEERING



BUSINESS PROCESS REENGINEERING

Business Definition	Business goals are identified within the context of key drivers: cost reduction, time reduction, quality improvement, and personnel development and empowerment.
Process Identification	Processes that are critical to achieving the goals defined in the business definition are identified.
Process Evaluation	The existing process is thoroughly analyzed and measured.
Specification and Design	Use-cases are prepared for each process that is to be redesigned.
Prototyping	A redesigned business process must be prototyped before it is fully integrated into the business.
Refinement and Instantiation	The business process is refined and then instantiated within a business system.

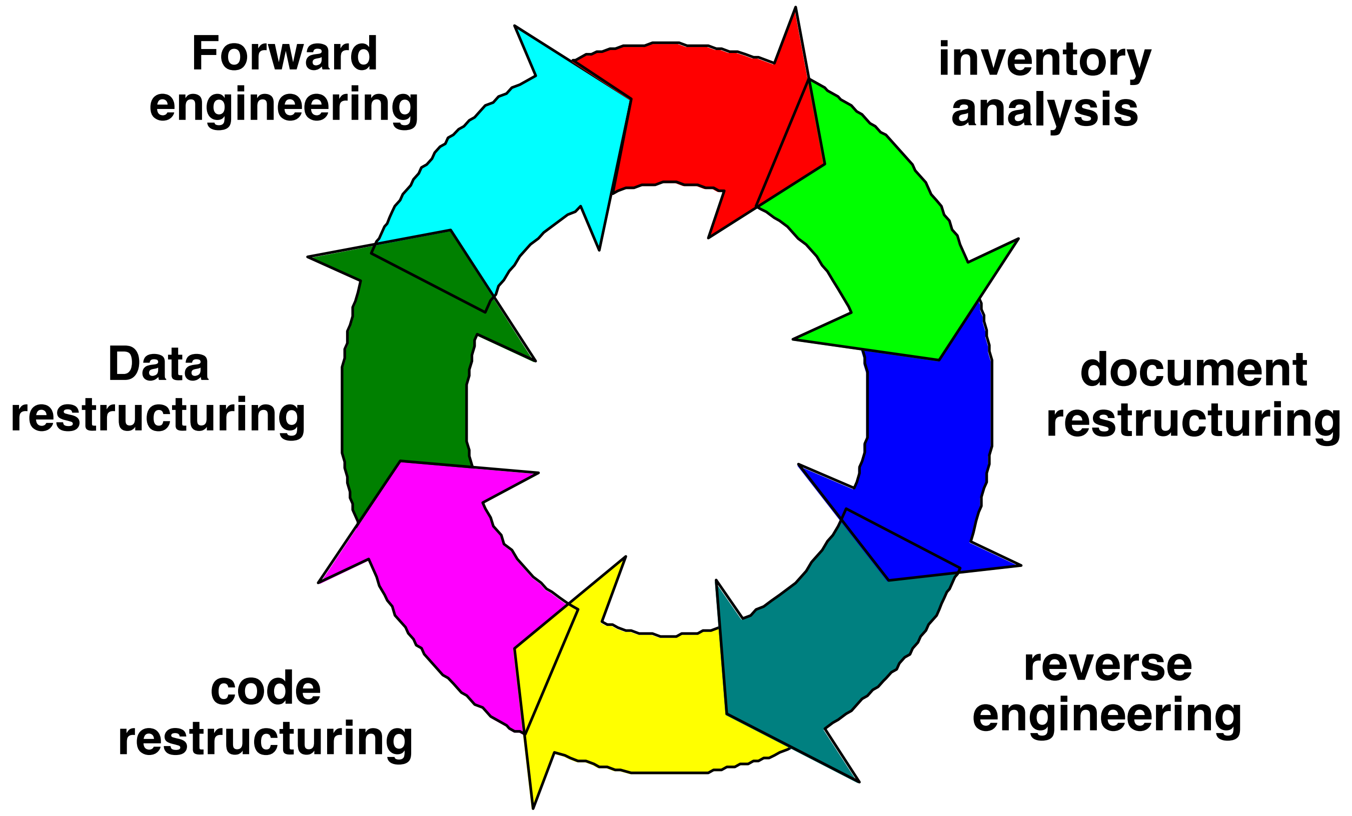


BUSINESS PROCESS REENGINEERING

BPR PRINCIPLES

Organize	Organize around outcomes, not tasks.
Right People	Have those who use the output of the process perform the process.
Gather Information	Incorporate information processing work into the real work that produces the raw information.
Resources	Treat geographically dispersed resources as though they were centralized.
Link Activities	Link parallel activities instead of integrated their results.
Decision	Put the decision point where the work is performed, and build control into the process.
Capture	Capture data once, at its source.

SOFTWARE REENGINEERING



INVENTORY ANALYSIS

Build a list that contains all applications that includes

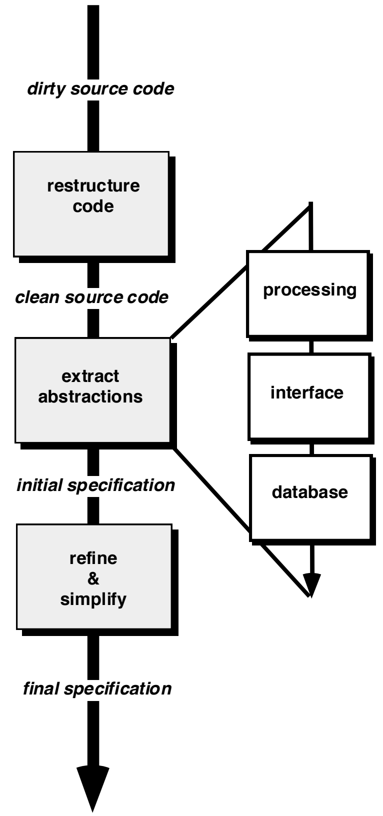
- name of the application
- year it was originally created
- number of substantive changes made to it
- total effort applied to make these changes
- date of last substantive change
- effort applied to make the last change
- system(s) in which it resides
- applications to which it interfaces

Analyze and prioritize to select candidates for reengineering

DOCUMENT RESTRUCTURING

- Weak documentation is the trademark of many legacy systems.
- Creating documentation is time consuming. If the system works, we'll live with what we have. In some cases, this is the correct approach.
- Documentation must be updated, but we have limited resources.
- Use a "document when touched" approach. It may not be necessary to fully redocument an application.
- The system is business critical and must be fully redocumented. Even in this case, an intelligent approach is to document only essential minimum.

REVERSE ENGINEERING



CODE RESTRUCTURING

Source code is analyzed using a restructuring tool.

Poorly design code segments are redesigned

Violations of structured programming constructs are noted and code is then restructured (this can be done automatically)

The resultant restructured code is reviewed and tested to ensure that no anomalies have been introduced

Internal code documentation is updated.

DATA RESTRUCTURING

- Unlike code restructuring, which occurs at a relatively low level of abstraction, data structuring is a full-scale reengineering activity
- In most cases, data restructuring begins with a reverse engineering activity.
- Current data architecture is dissected and necessary data models are defined.
- Data objects and attributes are identified, and existing data structures are reviewed for quality.
- When data structure is weak, the data are reengineered.
- Changes to the data will invariably result in either architectural or code-level changes.

FORWARD ENGINEERING

The cost to maintain one line of source code may be 20 to 40 times the cost of initial development of that line.

Redesign of the software architecture using modern design concepts, can greatly facilitate future maintenance.

Because the software already exists, development productivity should be much higher than average.

New requirements and the direction of change can be ascertained with greater ease.

A complete software configuration (documents, programs and data) will exist upon completion of preventive maintenance.

ECONOMICS OF REENGINEERING

The cost associated with continuing maintenance of a candidate application (i.e., reengineering is not performed) can be defined as

- Cmaint = [P3 - (P1 + P2)] x L

The costs associated with reengineering are defined using the following relationship:

- Creeng = [P6 - (P4 + P5) x (L - P8) - (P7 x P9)]

Using the costs presented in equations above, the overall benefit of reengineering can be computed as

- cost benefit = Creeng - Cmaint

ECONOMICS OF REENGINEERING

- Parameters are defined as below:
  - P1 = current annual maintenance cost for an application.
  - P2 = current annual operation cost for an application.
  - P3 = current annual business value of an application.
  - P4 = predicted annual maintenance cost after reengineering.
  - P5 = predicted annual operations cost after reengineering.
  - P6 = predicted annual business value after reengineering.
  - P7 = estimated reengineering costs.
  - P8 = estimated reengineering calendar time.
  - P9 = reengineering risk factor (P9 = 1.0 is nominal).
  - L = expected life of the system.

REFERENCE

- Roger Pressman, Software Engineering: A Practitioner's Approach, 8th edition, McGraw Hill, ISBN 0078022126