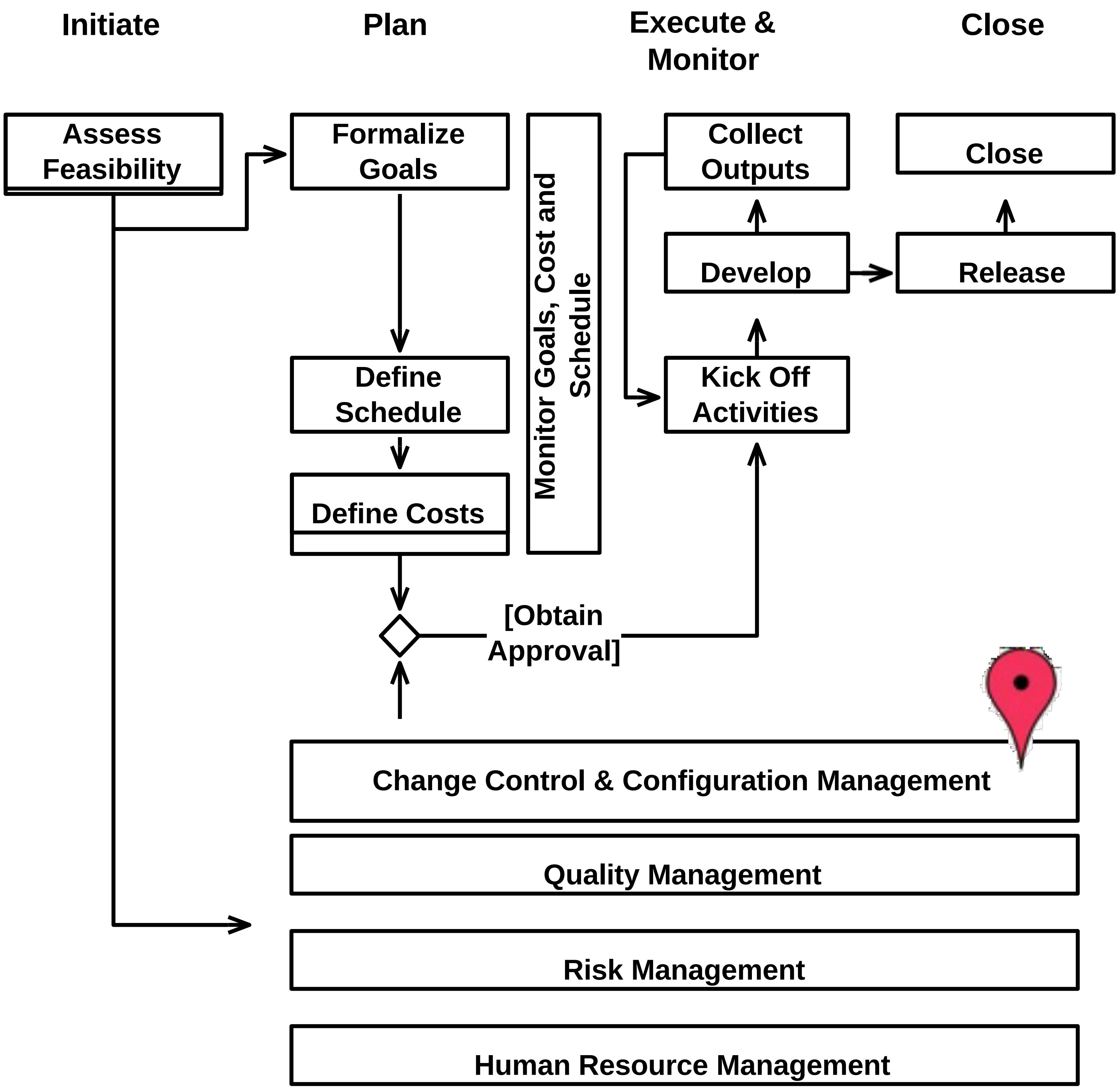
Software Project Management (9 – 20190423)

Mohammed Seyam
Assistant Professor
Information Systems Department
Faculty of Computers & Information
Mansoura University
seyam@mans.edu.eg
http://people.cs.vt.edu/seyam

Managing Changes, Risks, and Quality


Initiate
Plan
Execute & Monitor
Close
Assess Feasibility
Formalize Goals
Monitor Goals, Cost and Schedule
Collect Outputs
Close
Develop
Release
Define Schedule
Kick Off Activities
Define Costs
[Obtain Approval]
Change Control & Configuration Management
Quality Management
Risk Management
Human Resource Management

The Framework

- The scope document formalizes the goals of a project
- Ideally, once the goals are fixed, the project should move on to the design/implementation phase and achieve the project goals, through a progressive refinement
- Any deviation from such course of action is a perturbation (it changes goals, plans, costs, outputs, work to be performed, …)
- **Changes, however, are inevitable**
- The goal of a sound project management, therefore, is ensuring that the change process is properly managed

Fundamental Concepts

- **Change Control** is the set of practices to ensure request for changes are properly taken care of
- **Configuration Management** is the set of practices to ensure project outputs remain coherent over time

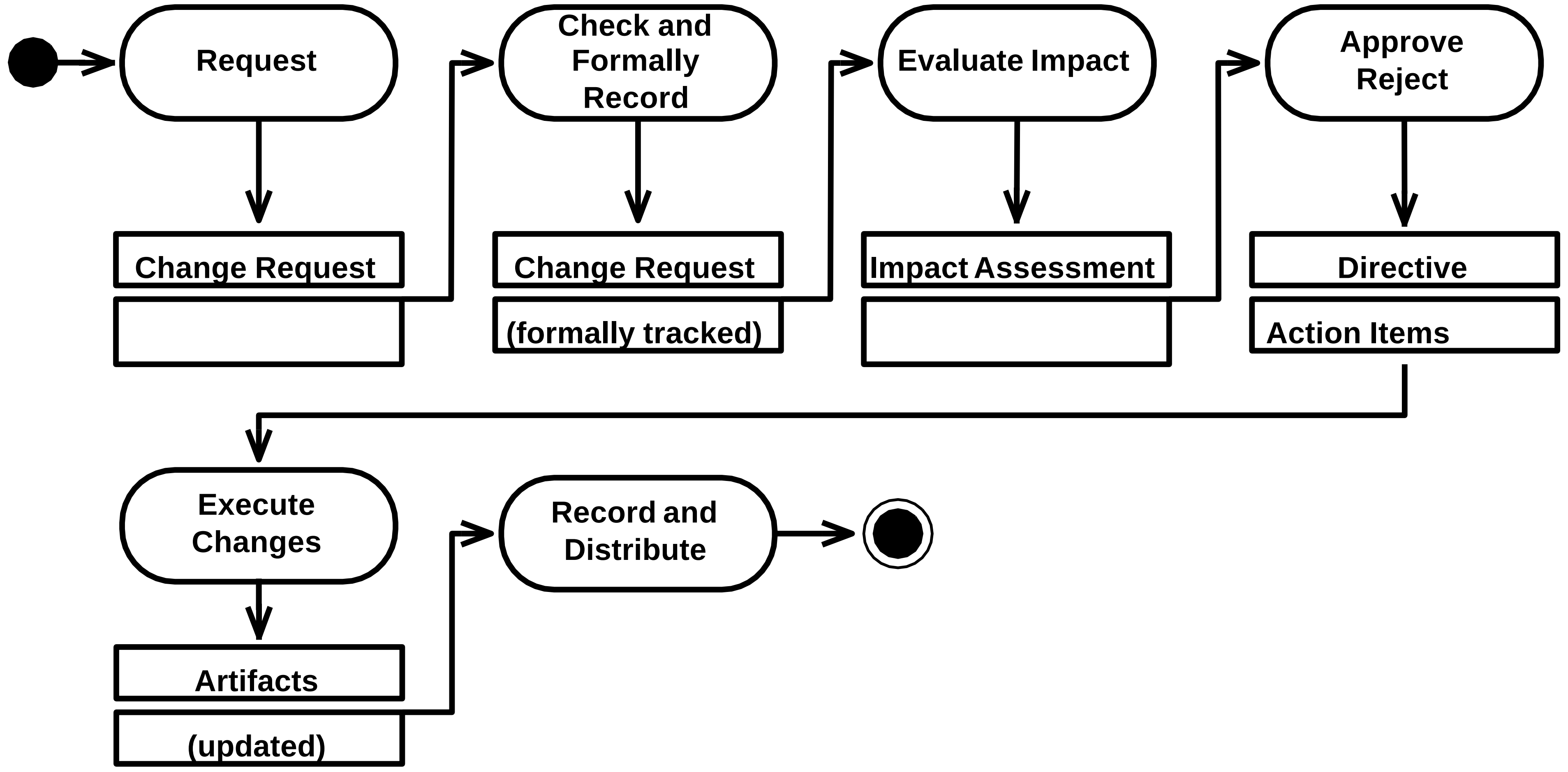
- Change Control and Configuration Management span over the lifecycle of the project outputs
- In software projects artifacts are extremely simple to change (e.g., editing a file)

Change Control

Causes of Request for Changes

- **Incompleteness or incoherencies** in the project requirements or in the description of work
- A **better comprehension of the system to be developed**
- A **technical opportunity**
- A **technical challenge**
- A **change in the external environment**
- **Non-compliance** of a project deliverable

A Change Control Process



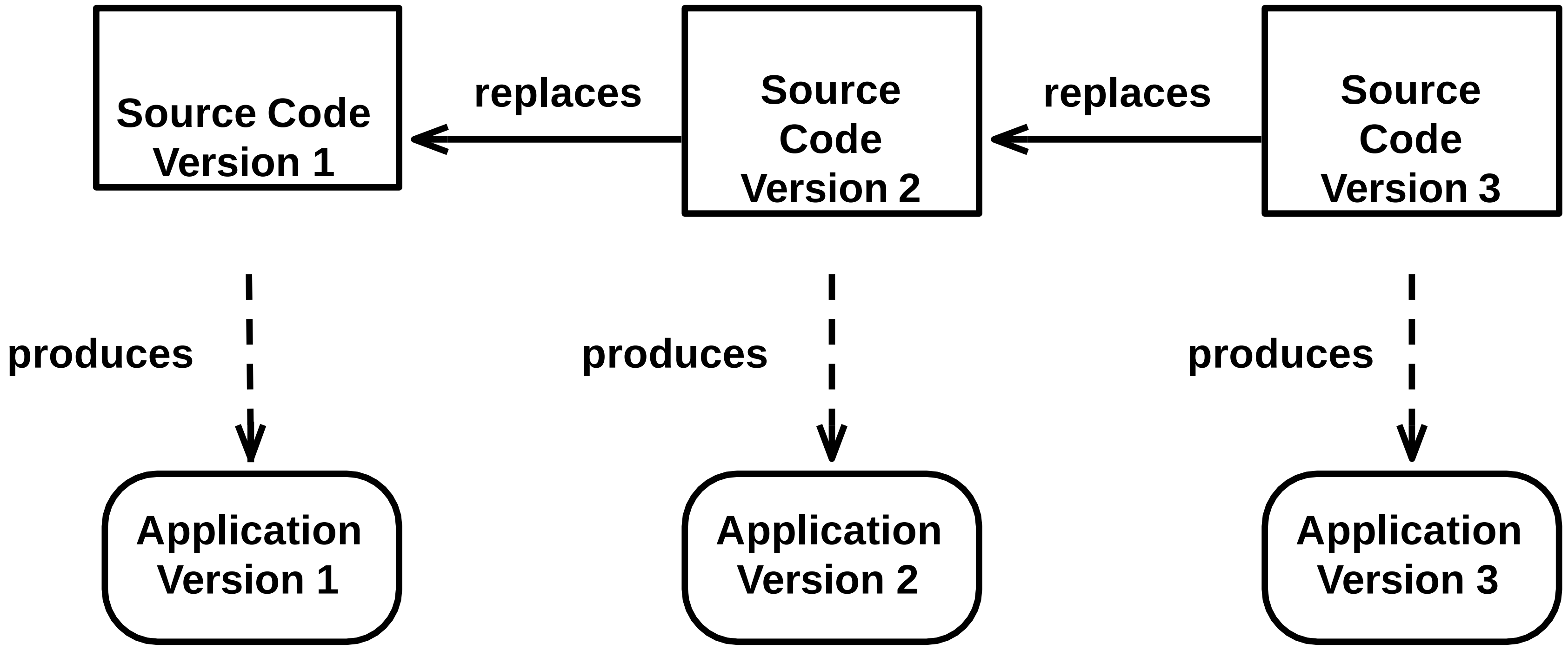
- It runs in parallel to the other PM activities throughout the project

Software Evolution Models

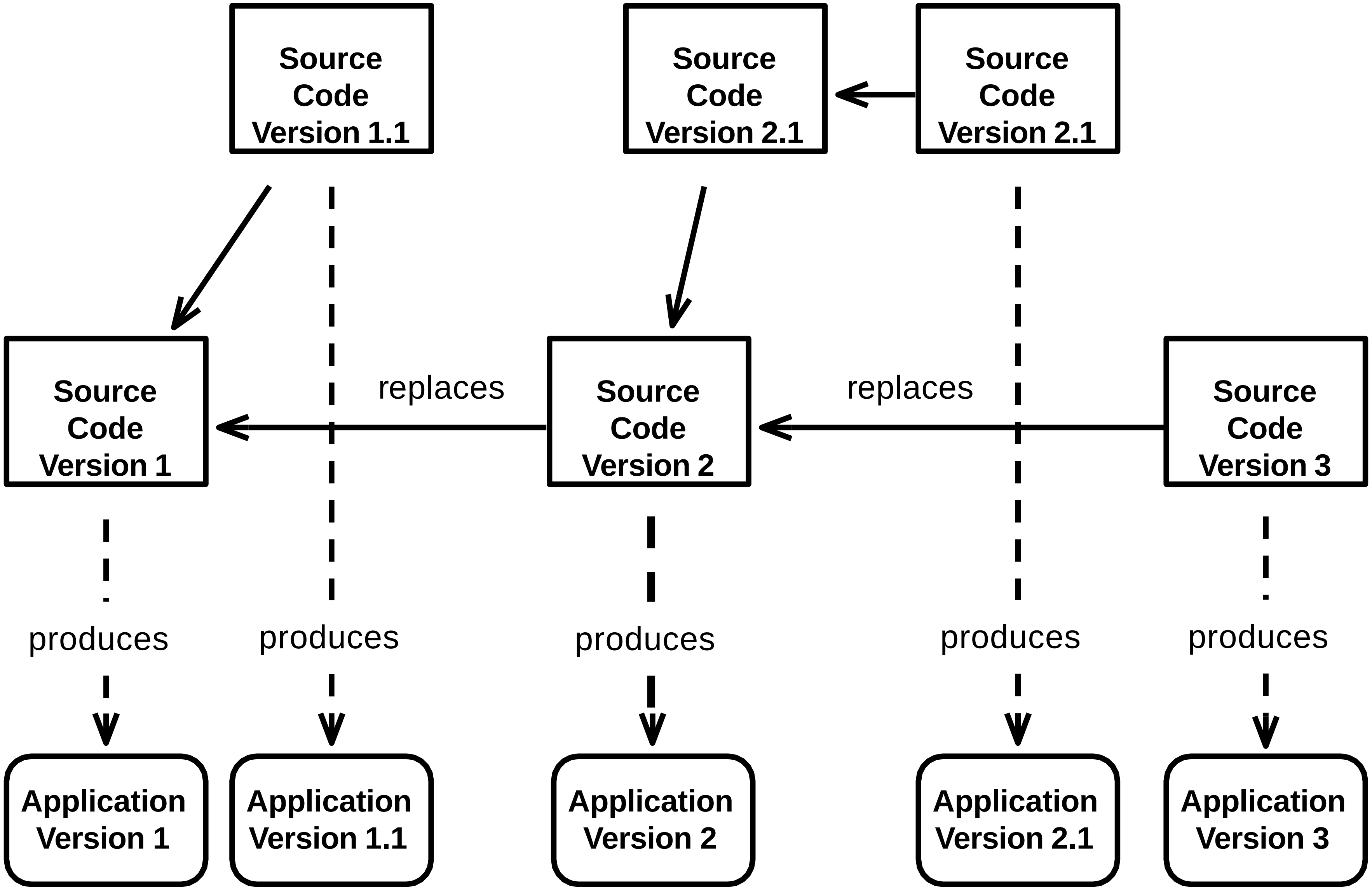
Software Development Models

- **Linear development:**
  - Only one version of an application is running at any given time (Example: one-offs; many web applications are one-offs)
- **Branching development:**
  - Various versions of an application are running at a given time

Linear Development Model



Branching Development Model



Configuration Management

Configuration Management

Configuration Management (CM) is a set of activities running in parallel to the development process, whose goal is **establishing** and **maintaining** system's coherency over time

- Part of the project management plan
- Helps define project standards and best practices

Configuration Management Main Goals

- Being able to **build a system** from a consistent set of **components**
- Being able to **retrieve a software component** when needed (consider: storage time, storage means)
- Being able to view the **history of changes** a system has undergone
- Being able to retrieve a **previous version** of a system

- Remark: closely related to the change management process

Some Examples

- A bug is reported by a user on a COTS software we have been selling for ten years.
- A client requests an enhancement to a one-off system we sold in 2005.
- We need to reproduce/understand an odd behavior of the control software of a space exploration probe which is now orbiting Jupiter

Steps and Tools: Establish Baseline

- The first step is “establishing what a product is”
- A good CM requires to:
  - Clearly identify the items which constitute a product
  - Identify the relationships among these items
  - Choose an appropriate identification and numbering scheme for versions
  - Take “snapshots”: baseline records

Steps and Tools: Manage Changes

- The second step is “maintaining coherency over time”
- A good CM process requires to:
  - Define the “baseline record” (the starting point)
  - Identify and approve requests for changes (see change control)
  - Formally record changes and history of each item
  - Maintaining old versions

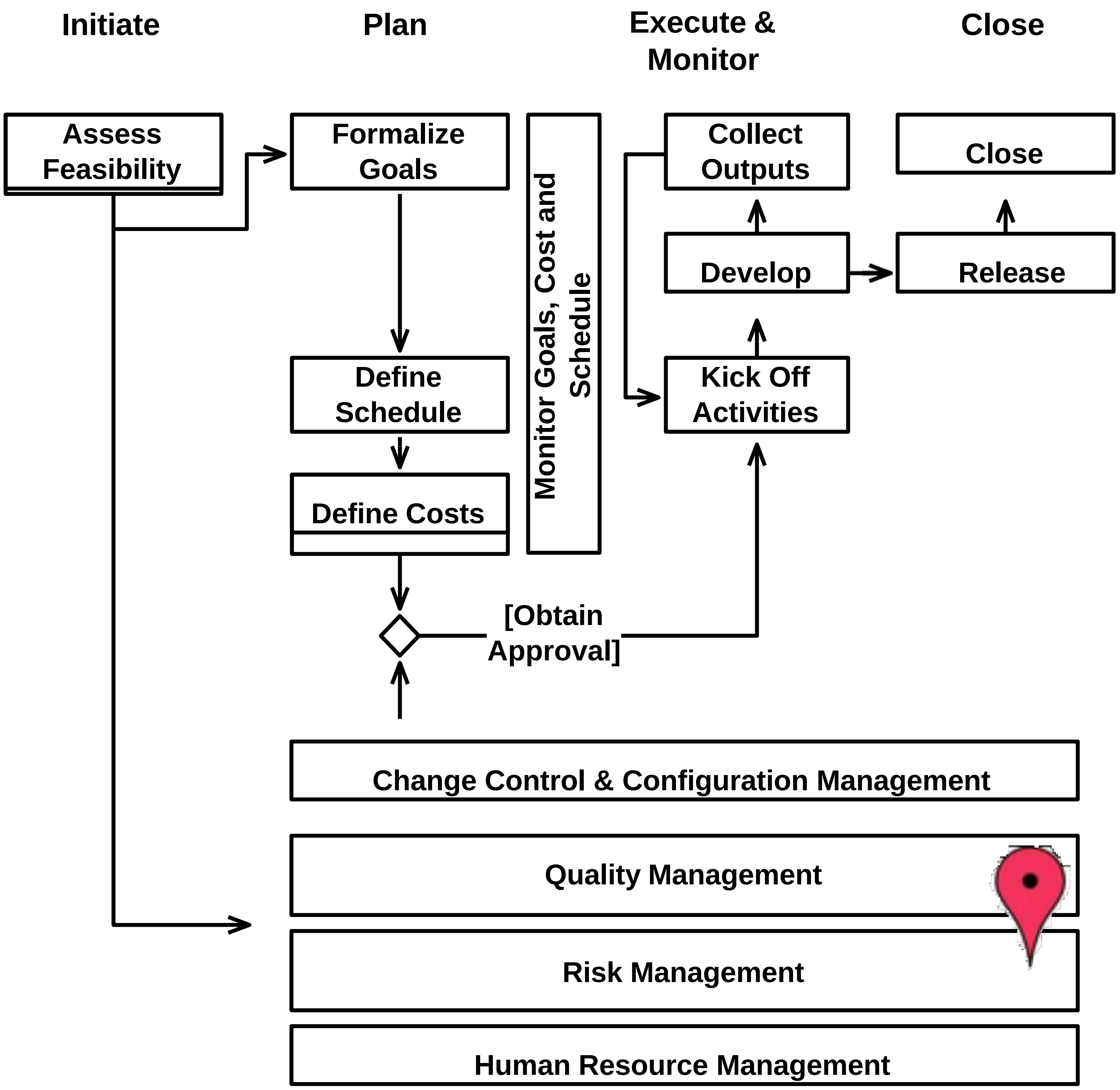
Steps and Tools: Considerations

- For software development, a **version control system** implements various of the functions described above
- Tools are not sufficient: an adequate process has to be in place

Version Control Systems: Main Concepts

- **Working version:** the file (or set of files we are currently editing)
- **Repository:** the storage where all versions of a file (or set of files) are kept together with additional information

Risk Management


Initiate
Plan
Execute & Monitor
Close
Assess Feasibility
Formalize Goals
Monitor Goals, Cost and Schedule
Collect Outputs
Close
Develop
Release
Define Schedule
Kick Off Activities
Define Costs
[Obtain Approval]
Change Control & Configuration Management
Quality Management
Risk Management
Human Resource Management

Motivations

- When we looked at **project selection** we just took into account financial data
- In the **scope management** document we emphasized the importance of making our goals achievable, i.e. the A in SMART ... however between achievable and achieved there is a big difference.
- In the **planning phase** we had to deal with various uncertainties (estimation) and tried to deal with them generically (e.g. time buffers)
- We stuck to one plan (the nominal plan), but the world is non-nominal: changes, both negative and positive, will occur!

Risk Management

Risk management collects techniques, know-how and processes to help identify, assess, manage, and monitor risks

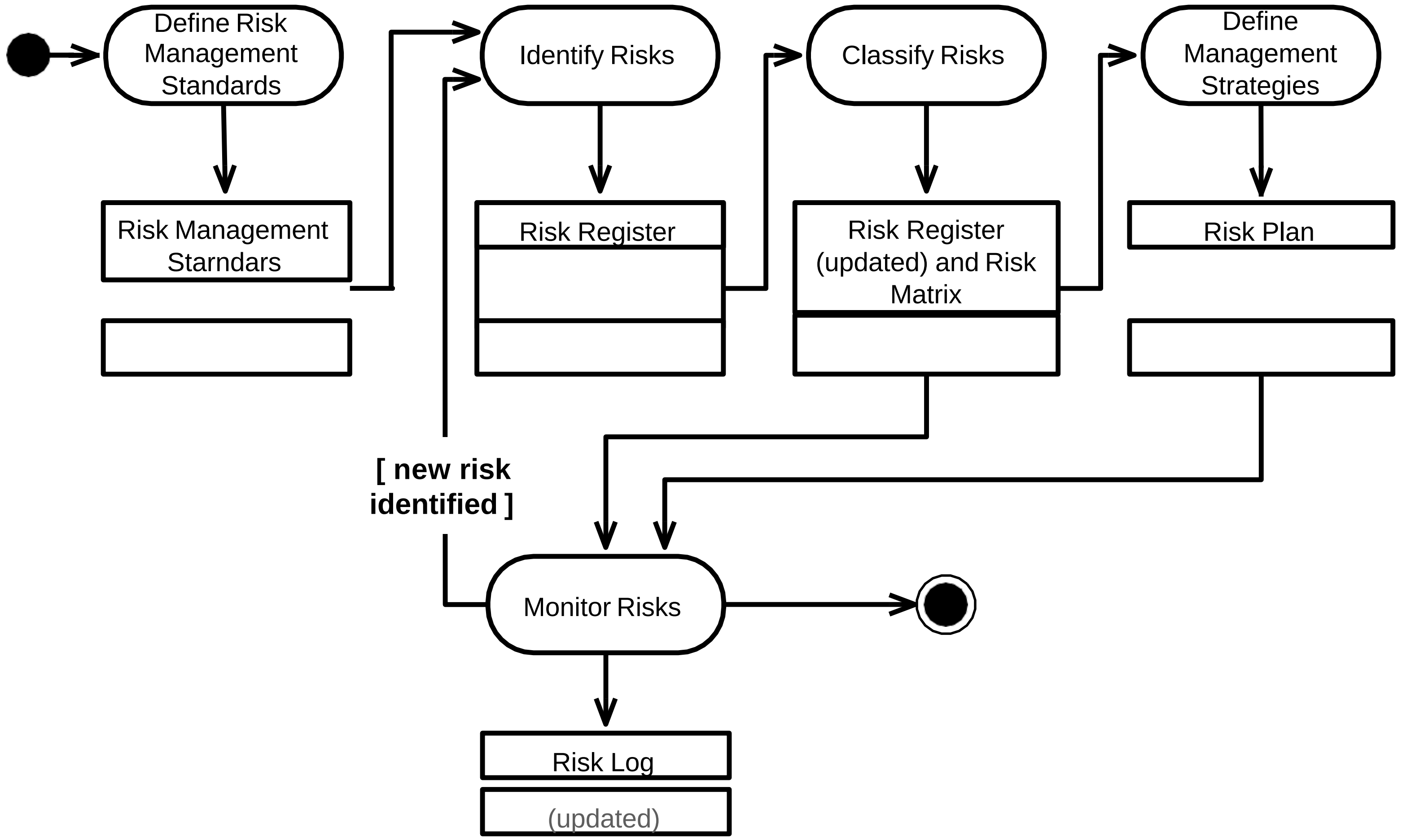
The objectives of Project Risk Management are to increase the probability and the impact of positive events and decrease the probability and impact of events adverse to the project.

Risk Management: Some Goals

- Understanding whether a project is worth taking
- Help refining the budget for the project
- Increase chances of ending the project successfully
- Increase chances of terminating the project as planned:
  - Within scope
  - Within quality
  - Within budget
  - On time

Risk Management and Project Management

The Risk Management Process


Define Risk Management Standards
Risk Management Starndars
Identify Risks
Risk Register
Classify Risks
Risk Register (updated) and Risk Matrix
Define Management Strategies
Risk Plan
[new risk identified]
Monitor Risks
Risk Log
(updated)

- It runs in parallel to the other PM activities throughout the project

Defining Risk Management Standards

Goal: describing how risk management will be structured and performed on the project.

- Output: a document (or set of documents and templates)
- Part of the project management plan
- Helps define project standards and best practices

Risk Identification

Goal: understanding what are the risk that could potentially influence the project and document their characteristics

- Risk identification is an iterative process (new risks may be identified as the project progresses; old risks may become “obsolete”)
- Output: Risk Register, basis for qualitative/quantitative risk analysis

Risk Identification and Classification

- Process (iterative):
  - Collect:
    - identify specific project risks
    - describe the risk
  - Analyze:
    - Identify the root causes (do not misinterpret effects as causes)
    - Define the risk category (impact) and probability
    - Identify other useful characteristics:
      - When it can occur or frequency of occurrence
      - How it manifests
- Output:
  - Risk Register

Risk Identification Techniques

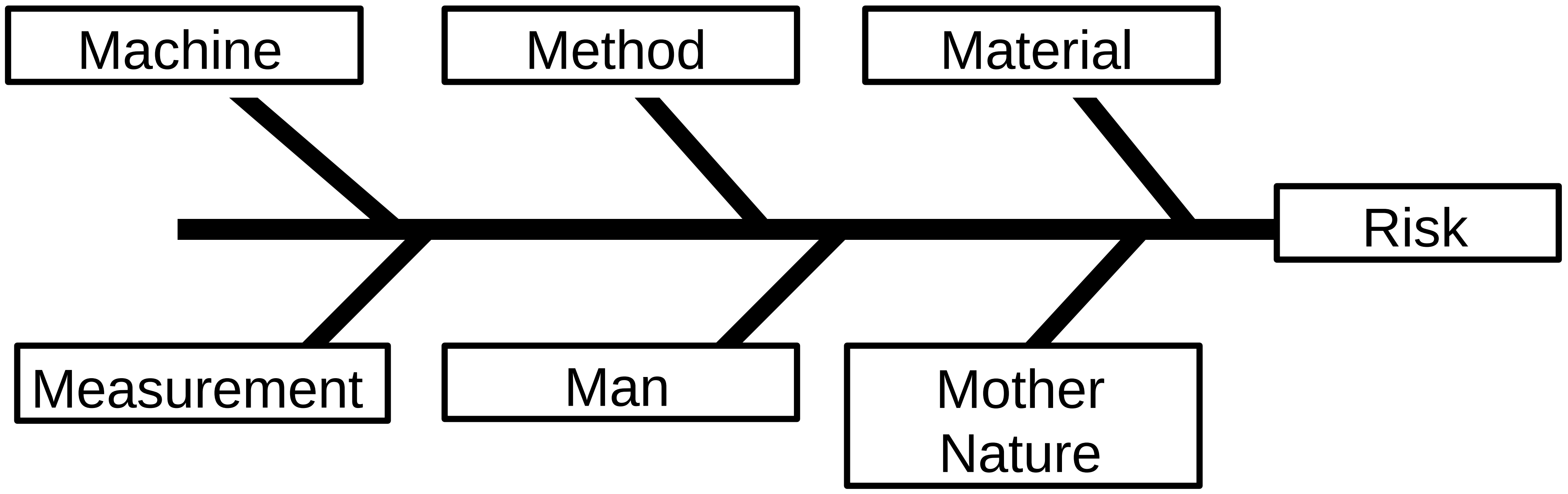
- Meetings
- Document Analysis
- Risk Breakdown Structures, Checklists, Templates
- Analogy

Boehm’s Top Ten Causes for Project Failures

- Boehm developed a list of the ten most common causes for projects to fail
- Some of the causes mentioned in the list can be used as starting point to identify the risks applicable to a project at hand
- Risks include:
  - Personnel or subcontractors Shortfall
  - Unrealistic schedule and budget
  - Developing the wrong software functions/user interface
  - Gold plating (getting priorities wrong)
  - Ineffective change control
  - Technical risks

Root Cause Analysis Techniques

- Cause-Effect Diagram (Ishikawa)
- Fault Trees/Failure Modes and Effect Analysis



Fishbone Diagrams: Some starting points

- **The 6 M's:**
  - Machine, Method, Materials, Measurement, Man and Mother Nature (Environment)
    (recommended for manufacturing industry).
- **The 8 P's:**
  - Price, Promotion, People, Processes, Place / Plant, Policies, Procedures & Product (or Service)
    (recommended for administration and service industry).
- **The 4 S's:**
  - Surroundings, Suppliers, Systems, Skills
    (recommended for service industry).

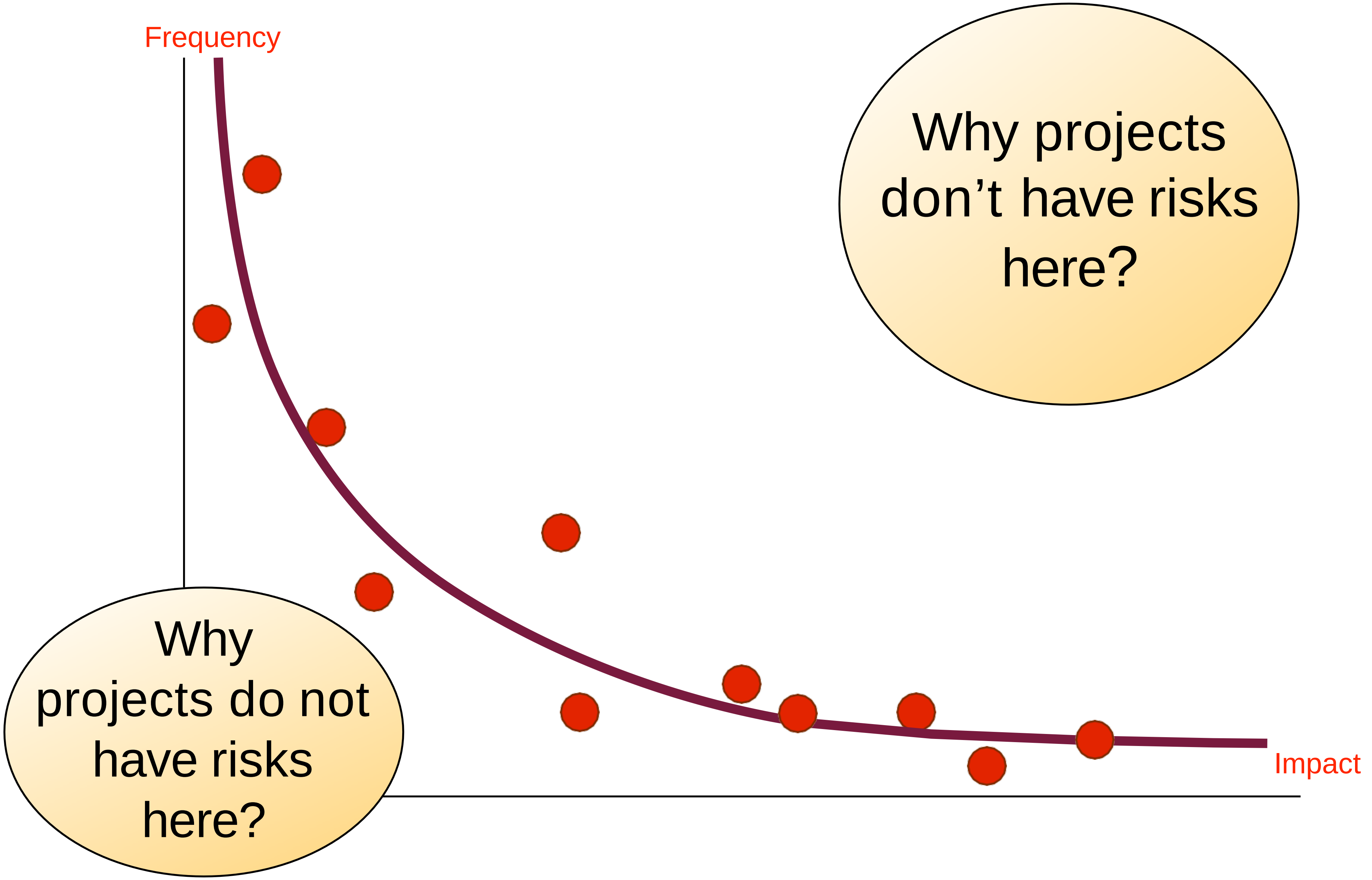
Risk Assessment and Risk Management Strategies

Risk Assessment

Goal: prioritize risks according to their impact and likeness on the project

- Output: a prioritized list of risks (priority defined according to probability and impact)
- Information on whether a project is worth taking
- Information about what risks must be monitored

Probability/Impact



Techniques

- **Qualitative risk analysis**
  - Simpler
  - Can be used when no precise information about probabilities of risk is available
- **Quantitative risk analysis**
  - More systematic
  - Suitable for mathematical analysis
  - Provide figures on the (economical) impact of risks

Risk Management Strategies

a.k.a. Risk Response Planning: how do we take care and exploit risks

Risk Management Strategy

Goal: find a treatment for the unacceptable risks and decide the strategies to apply for the remaining risks, should they occur during the project

- Output: a plan with only acceptable risks
- A contingency plan for each remaining significant risk

The Scenario

- RED: Require special treatment (or drop the project)
- ORANGE: Need close monitoring
- GREEN: LOW: Standard in a project... nuisances

	Negligible	Low	Moderate	Severe	Catastrophic
Very High	🟩 R1	🟥	🟥	🟥	🟥 R5
High	🟩	🟧	🟥 R2	🟥 R6, R7, R8	🟥
Moderate	🟩	🟧 R3	🟧	🟥	🟥
Low	🟩	🟩	🟧	🟧 R4	🟥
Very Low	🟩	🟩 R9, R10	🟩	🟧	🟥

Strategies: Menaces

- **Avoid**
  - Change the plan to eliminate the threat (increase time, relax objectives, take corrective actions - increase time to do requirements)
- **Transfer**
  - Shift the negative outcome to a third party. It transfers responsibility, it does not eliminate the risk (insurance, contracts to transfer liability… they require to pay you a price)
- **Mitigate**
  - Reduce probability or impact (often better than trying and repair the damage; prototyping)

Strategies: Opportunities

- **Exploit**
  - Eliminate uncertainty relate to the occurrence of the opportunity (e.g. assign more talented people, provide better quality)
- **Share**
  - Allocate responsibility of exploitation to a third party (joint-ventures, partnerships, …)
- **Enhance**
  - Modify the size of an opportunity by increasing probability and/or positive impact

Strategy: common

- ***Accept***
  - **Passive:** just let the team deal with the risks
  - **Active:** provide some buffer (time, money, …)

Why?

... Low impact or probability

... Simpler to deal with the risk, if it occurs than planning a response in advance

The Risk Register

- The most common tool to list and manage risks is a spreadsheet
- One row per risk
- Each risk characterized by:
  - ID, Title, Description
  - Risk Category (if you are inclined to classifications)
  - Probability, Impact and, possibly, Score (PxI)
  - Root cause
  - Time-frame
  - Monitoring modalities (periodicity, person, reporting)
  - Status (active, occurred, inactive)

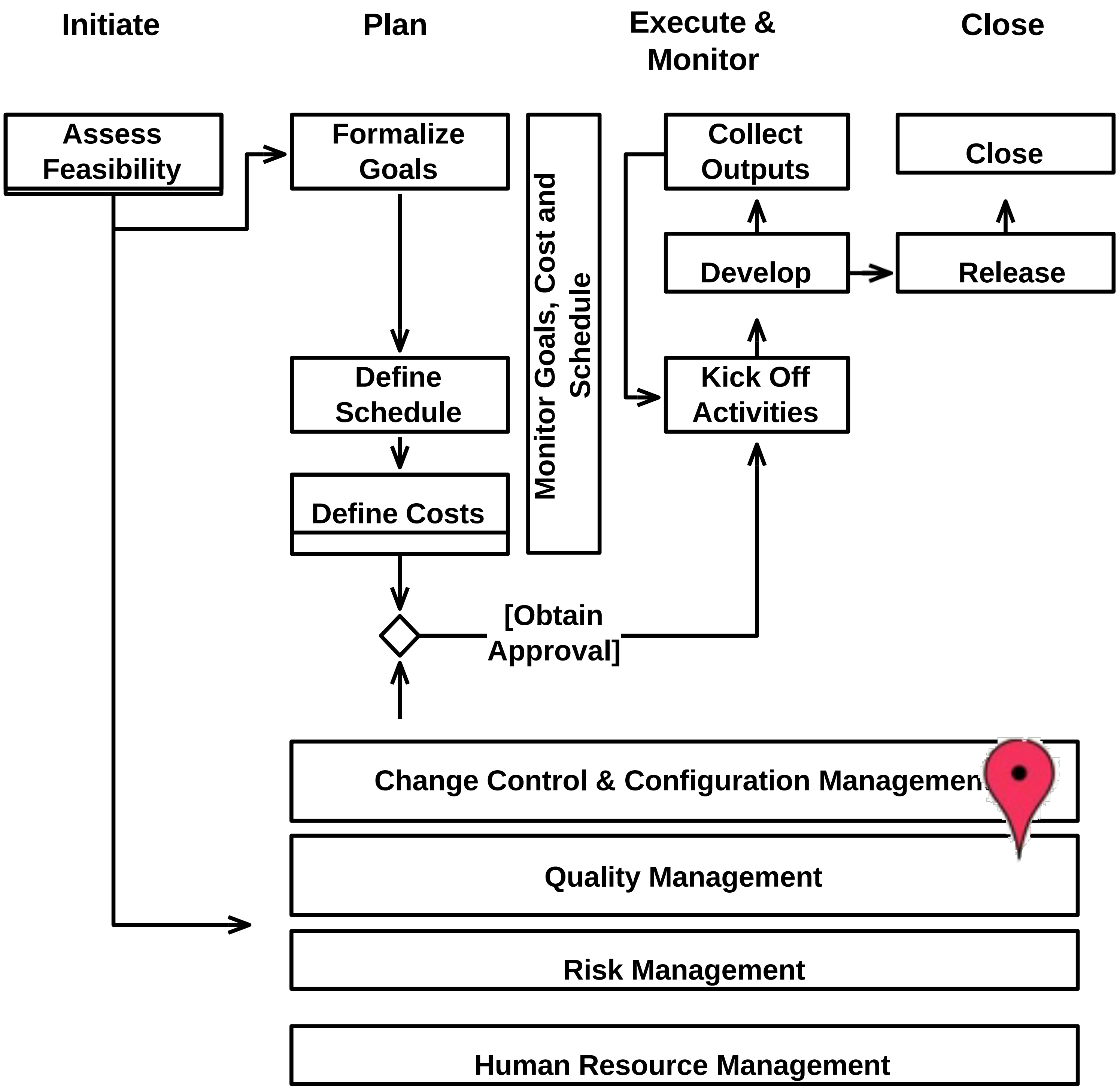
Risk Monitoring and Control

a.k.a. Risk Response Planning: how do we take care and exploit risks

Risk Monitoring and Control

- Input:
  - The risk register
- Process
  - Analyze deviations from the nominal plan
  - Identify causes
  - Evaluate corrective actions
  - Modify current plan
- Mind:
  - Planned risks must be dealt with as above (use contingency plans)
  - Unplanned risks require the full process!

Quality Management


Initiate
Plan
Execute & Monitor
Close
Assess Feasibility
Formalize Goals
Monitor Goals, Cost and Schedule
Collect Outputs
Close
Develop
Release
Define Schedule
Kick Off Activities
Define Costs
[Obtain Approval]
Change Control & Configuration Management
Quality Management
Risk Management
Human Resource Management

Software Quality Assurance

Software Quality Assurance

Software quality assurance is the planned and systematic application of activities to ensure **conformance** of *software life cycle processes* and *products* to requirements, standards, and procedures

Quality Assurance Process

- **Quality planning,** which identifies the relevant standard and practices and the way to implement them
- **Quality assurance,** which focuses on ensuring that the project applies and follows the quality standards identified at the previous step
- **Quality control,** which ensures that the products respect the quality standards identified during the planning phase

Quality Planning

Quality Planning

- Goal: ensure the goals of quality management are met in a project
- Means:
  - Identification of constraints and **quality goals** in scope
  - Identification of **standards** and procedures to be applied
  - Identification of **techniques** to be applied
  - Allocation of **resources** (time, people, budget) to quality assurance activities
  - Roles and **responsibilities**
- Output: quality assurance planning document

Comments

- Quality needs to be balanced with the other project constraints (e.g. time and costs)
- Not all systems are equally critical: NASA, for instance distinguishes eight different classes of software systems
- The quality assurance team should be **independent** from the development team

Quality Assurance & Quality Control

Quality Assurance

- **Goal: ensure that the project applies and follows the quality standards**
- Main tool: quality audits
- Triggers: time, milestones, or critical events in the project (according to the quality plan)
- Quality audits include
  - Inspections
  - Reviews
  - Walkthroughs
- Output:
  - Main findings and recommendations

Signs of Troublesome Projects

- According to NASA signs of troublesome projects include:
  - Frequent changes in milestones
  - Unexplained fluctuations in personnel
  - Continued delays in software delivery
  - Unreasonable number of non conformance reports or change requests.

Quality Control

- **Goal: ensure that the the <u>products</u> respect the quality standards identified during the planning phase**
- Main tools:
  - Inspections
  - Analyses
  - Testing
- Triggers: milestones or critical events in the project (according to the quality plan)
- Output:
  - List of non-conformance reports

Quality Control

- Quality control of software systems is extremely difficult, because:
  - of the enormous number of states a software system can be in (exhaustive testing is impractical/impossible)
  - the operating environment is unpredictable
  - discontinuity: little changes in inputs can cause enormous changes in outputs
  - non functional requirements can be difficult to assess (consider, e.g., maintainability, usability)
  - test automation can be difficult or very costly (consider, e.g., testing a GUI)
  - today's systems are composed by using different technologies (e.g., HTML/CSS, Javascript, PHP, WebServer, OS)

Questions

?