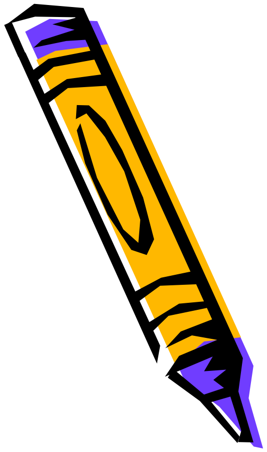
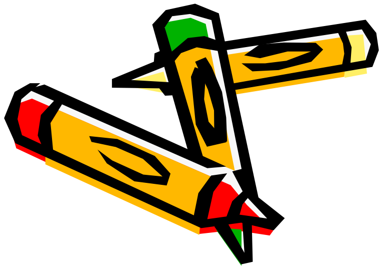
# Introduction to Software Process Engineering Metamodel

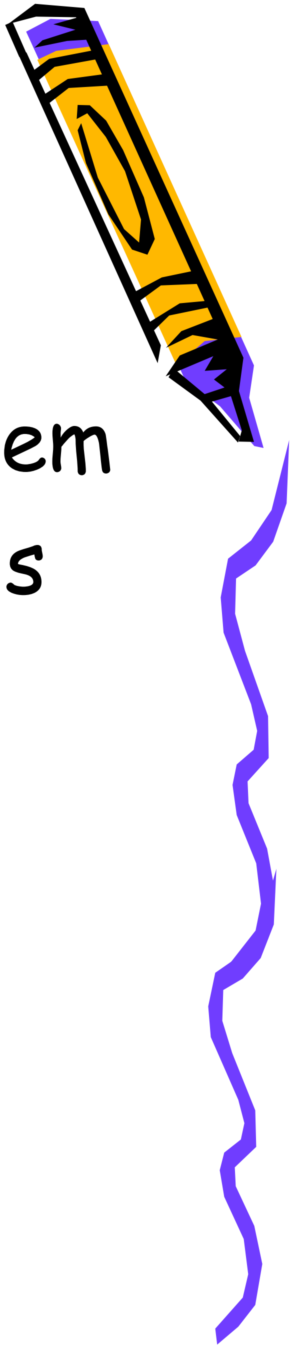
Dmitry Malenko
dmalenko@acm.org



# Agenda

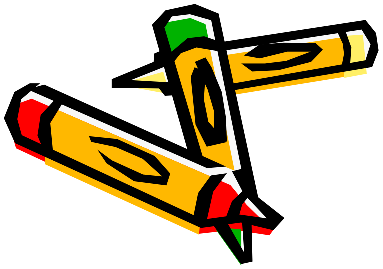
- Modeling with UML
- What is SPEM?
- Modeling software processes using SPEM
- SPEM in Software Engineering



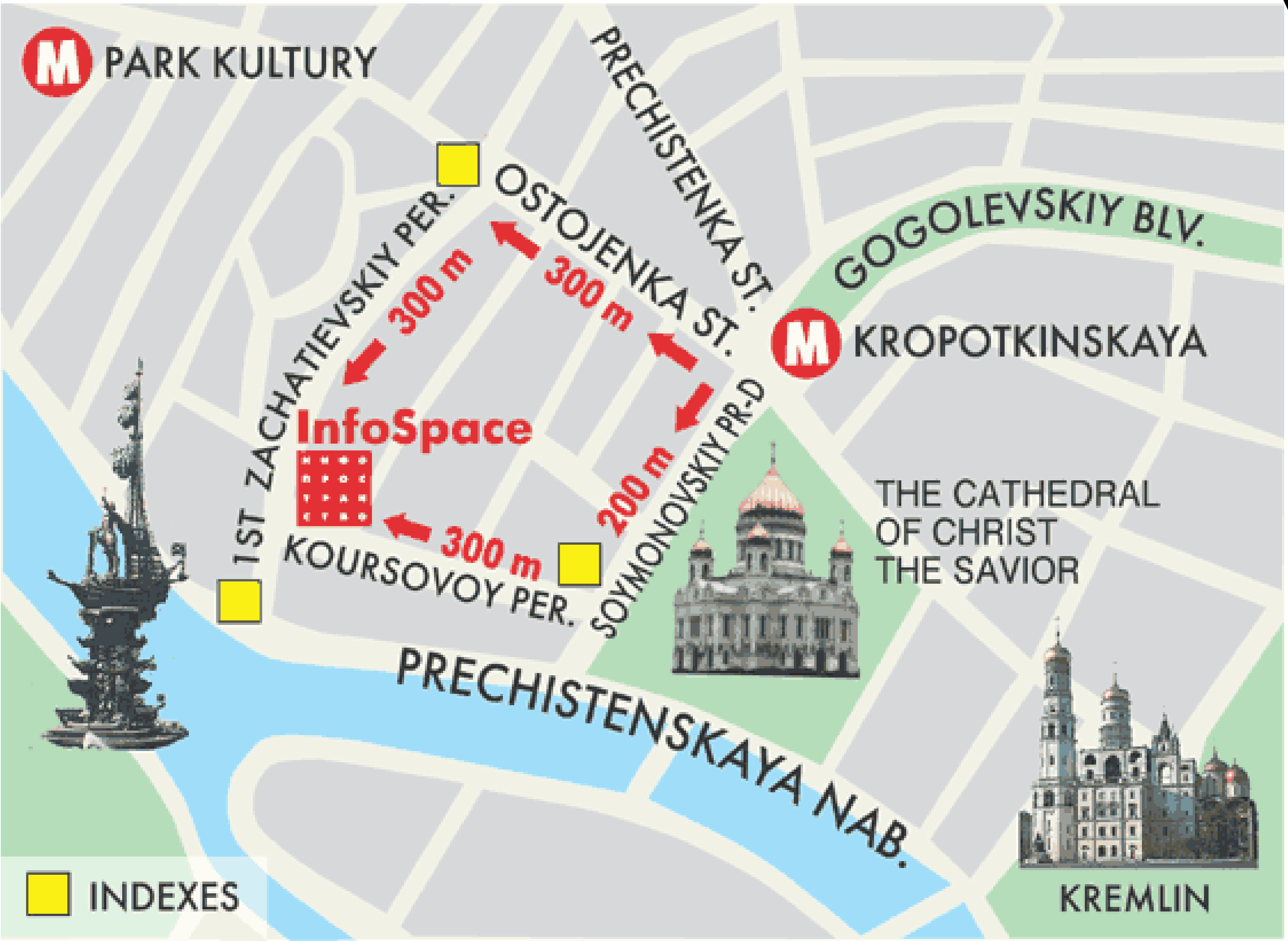


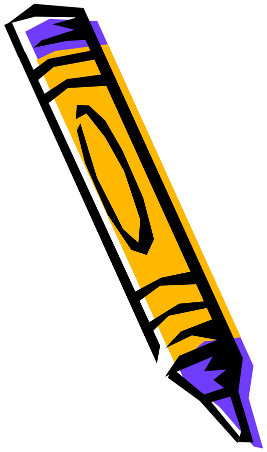
# Why modeling?

- Helps better understand the problem
- Abstracts away unnecessary details
- Allows to envision the solution
- Communicates the decisions made



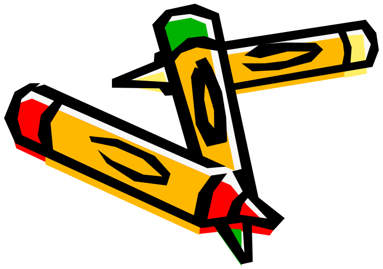
# What is this?


PARK KULTURY
PRECHISTENKA ST.
OSTOJENKA ST.
GOGOLEVSKIY BLV.
KROPOTKINSKAYA
1ST ZACHATIEVSKIY PER.
300 m
300 m
InfoSpace
200 m
SOYMONOVSKIY PR-D
300 m
KOURSOVOY PER.
THE CATHEDRAL OF CHRIST THE SAVIOR
PRECHISTENSKAYA NAB.
KREMLIN
INDEXES



# Why UML?

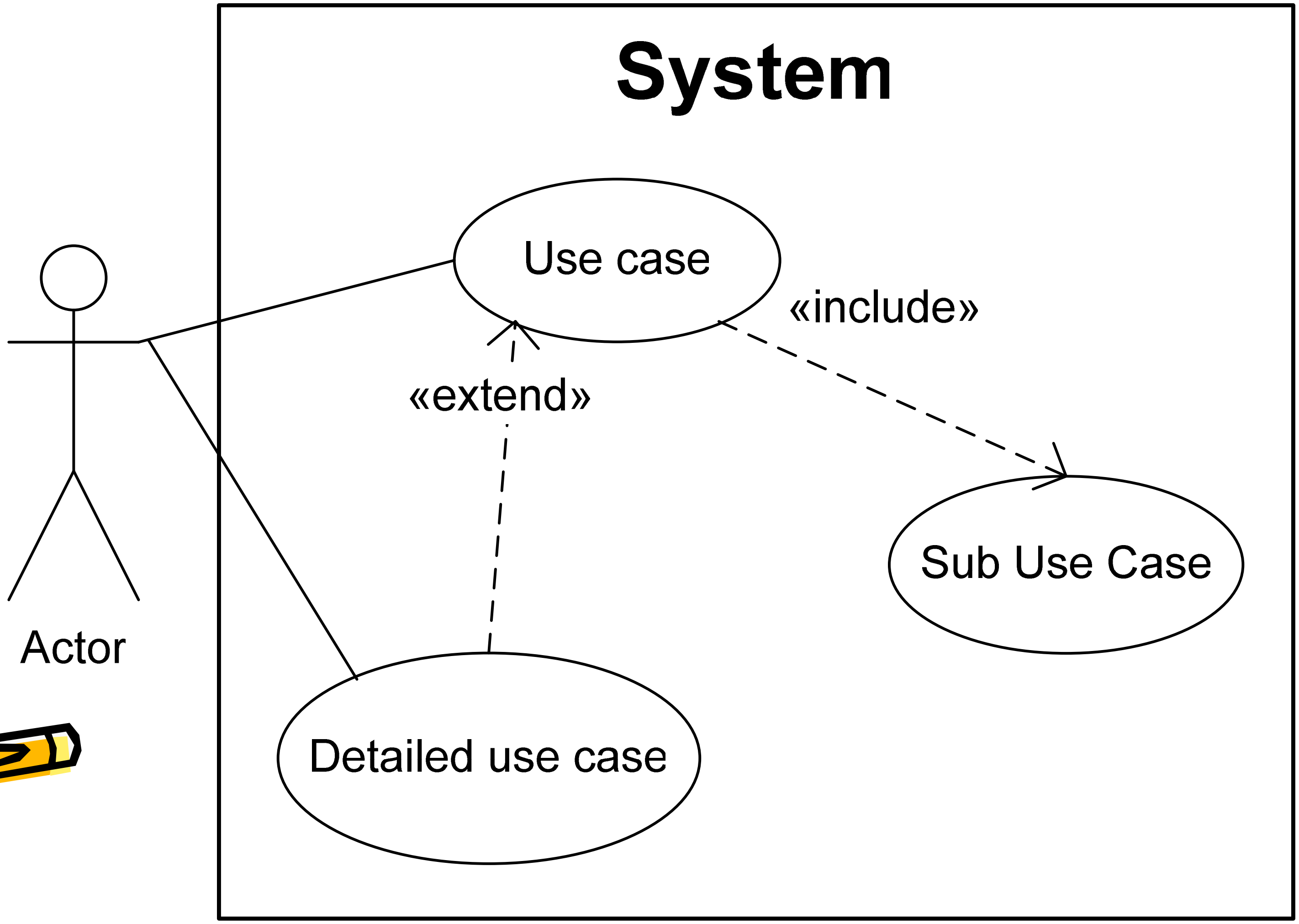
- Unified Modeling Language
- For modeling software systems, business modeling and other non-software systems
- Unifies
  - modeling languages used before
  - perspectives among different kind of phases, development phases



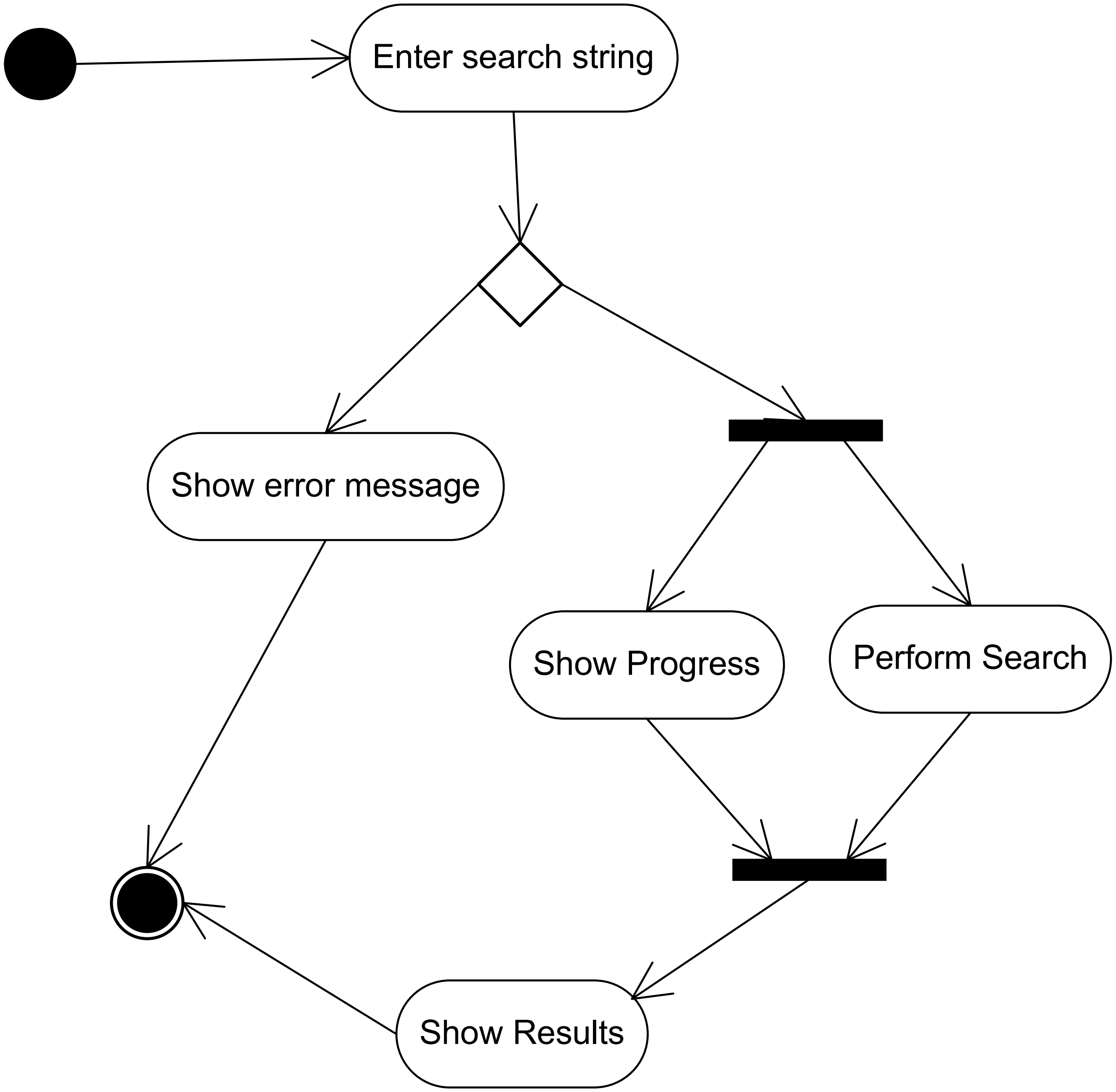
# Views in UML

- Use case view
  - External view of the system
- Logical view
  - Structure and behavior of the system
- Implementation view
  - Details of implementation

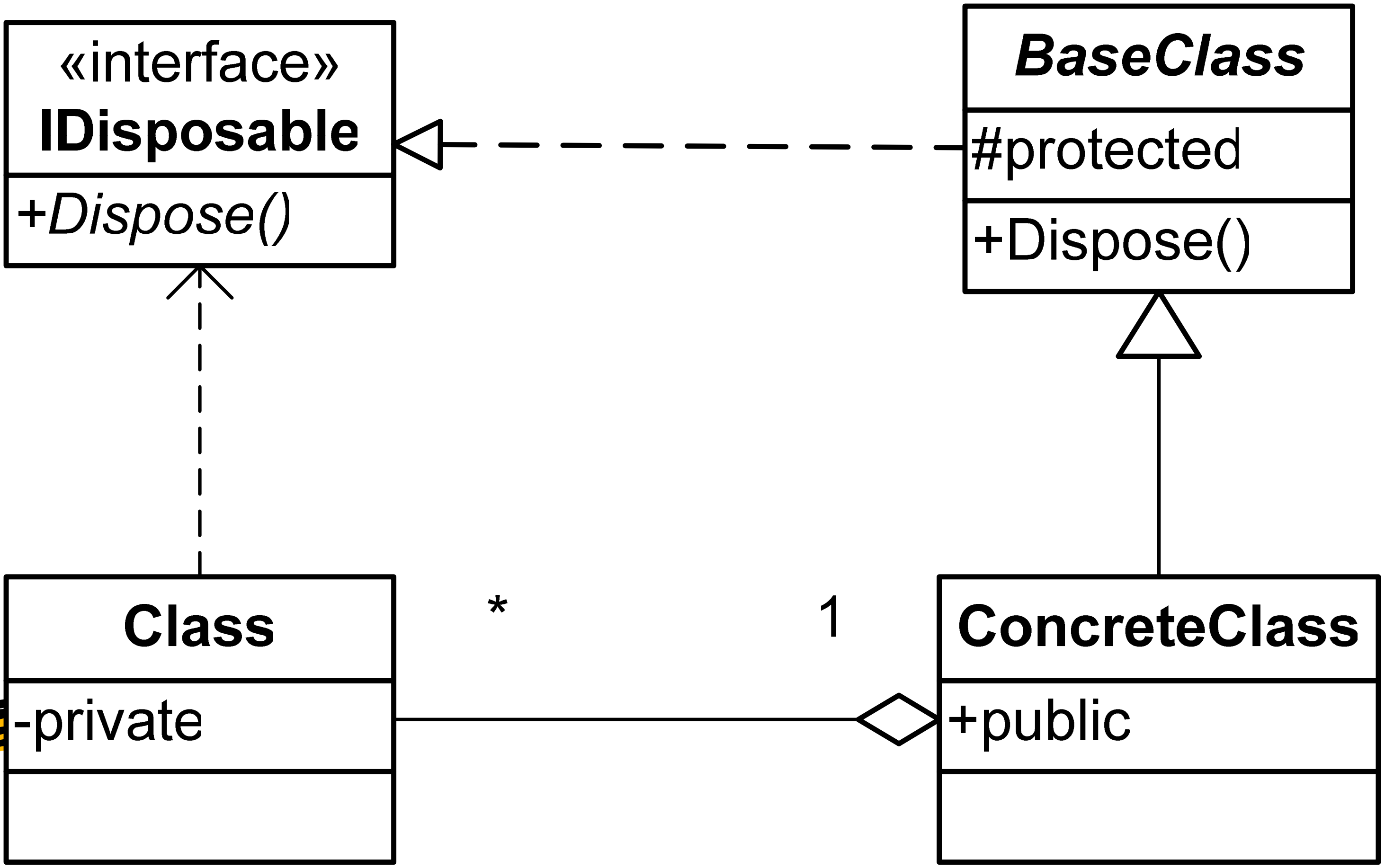
# Use case diagram



# Activity diagram



# Class diagram



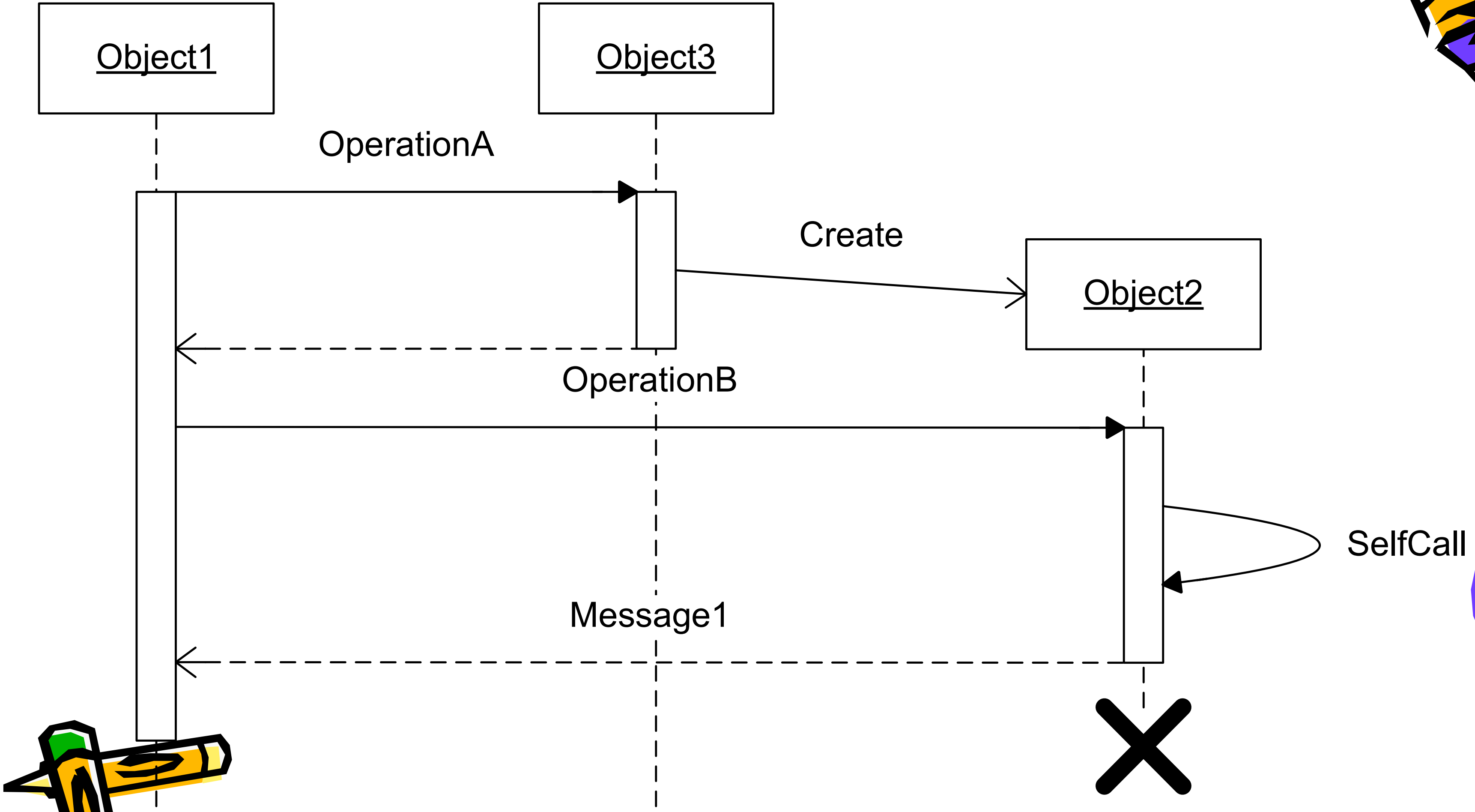
«interface»
**IDisposable**
+*Dispose()*

***BaseClass***
#protected
+Dispose()

**Class**
-private

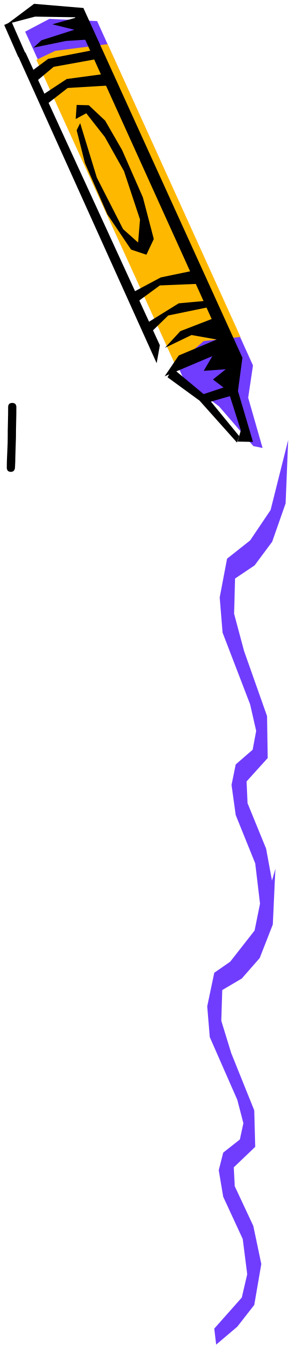
**ConcreteClass**
+public

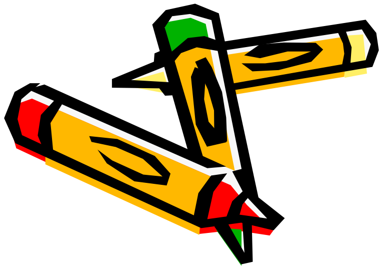
# Sequence diagram



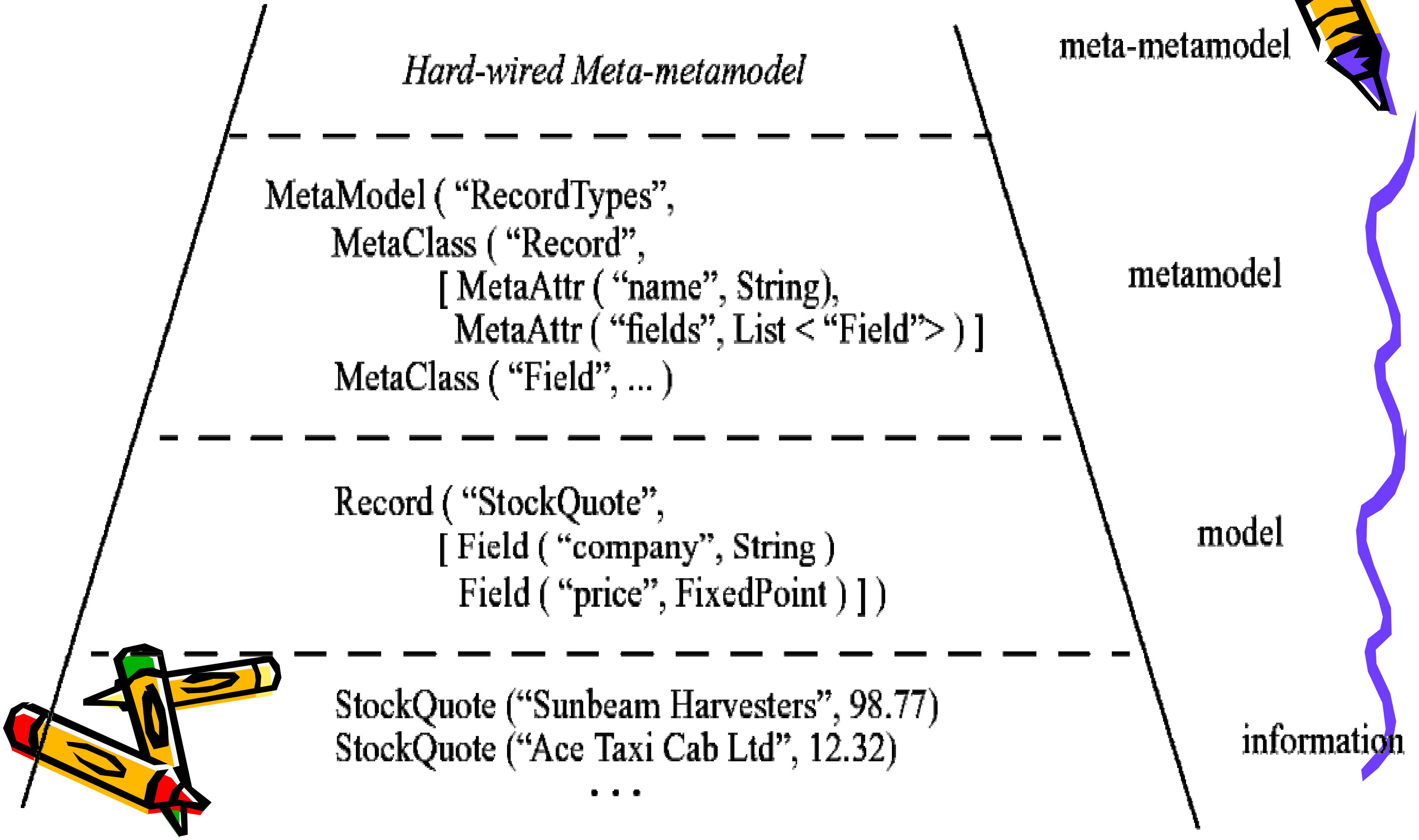
# UML Metamodel

- UML is described by its metamodel
- Metamodel is definition of UML in UML
- UML metamodel builds semantic foundation of UML





# Levels of metamodels



meta-metamodel

*Hard-wired Meta-metamodel*

metamodel

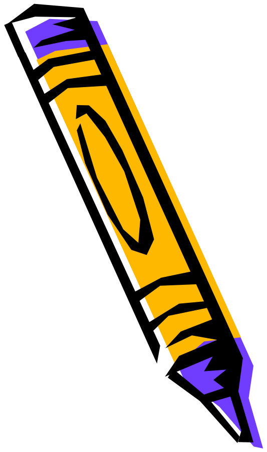
```
MetaModel ( "RecordTypes",
     MetaClass ( "Record",
             [ MetaAttr ( "name", String),
               MetaAttr ( "fields", List < "Field"> ) ]
     MetaClass ( "Field", ... )
```

model

```
Record ( "StockQuote",
        [ Field ( "company", String )
          Field ( "price", FixedPoint ) ] )
```

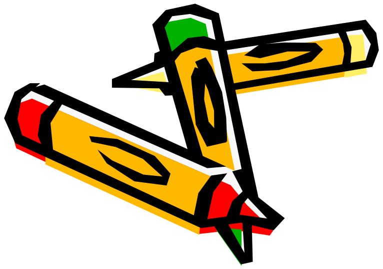
information

```
StockQuote ("Sunbeam Harvesters", 98.77)
StockQuote ("Ace Taxi Cab Ltd", 12.32)
                 . . .
```



# UML Profiles

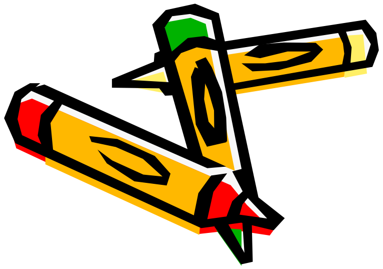
- Domain specific extensions of UML
- Defined through standard extension mechanisms of UML
- Some are standardized by OMG
  - UML Profile for CORBA
  - UML Testing Profile
  - Software Process Engineering Metamodel
  - …

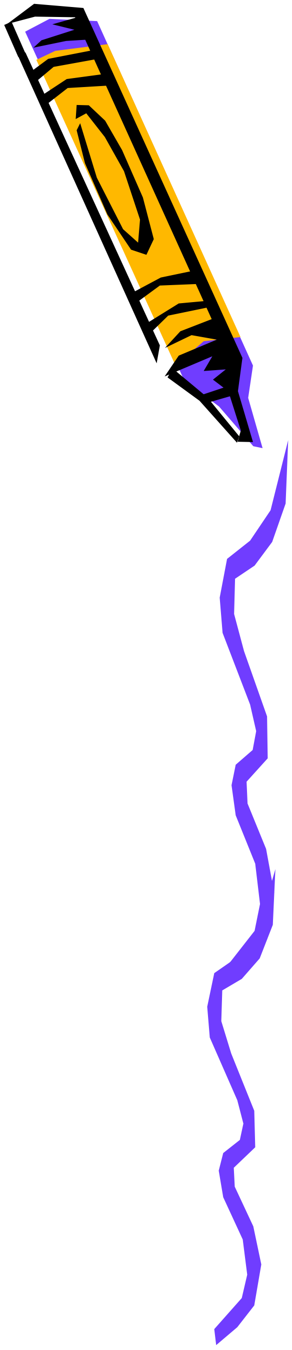


# Software Process Engineering Metamodel

This metamodel is used to describe a concrete software development process or a family of related software development processes.

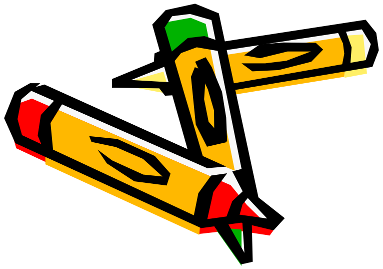
Software Process Engineering Metamodel Specification

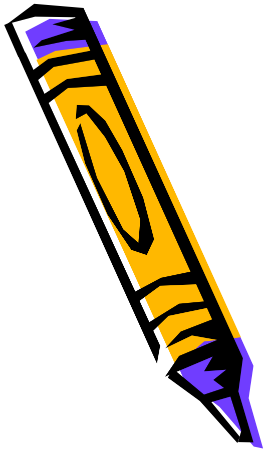




# Software Process

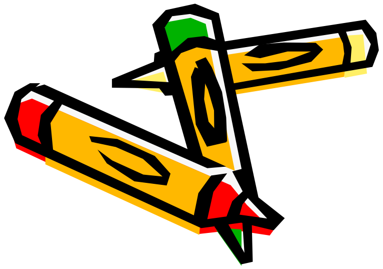
- Is not merely a workflow!!!
- It defines
  - roles and responsibilities
  - qualifications
  - artifacts



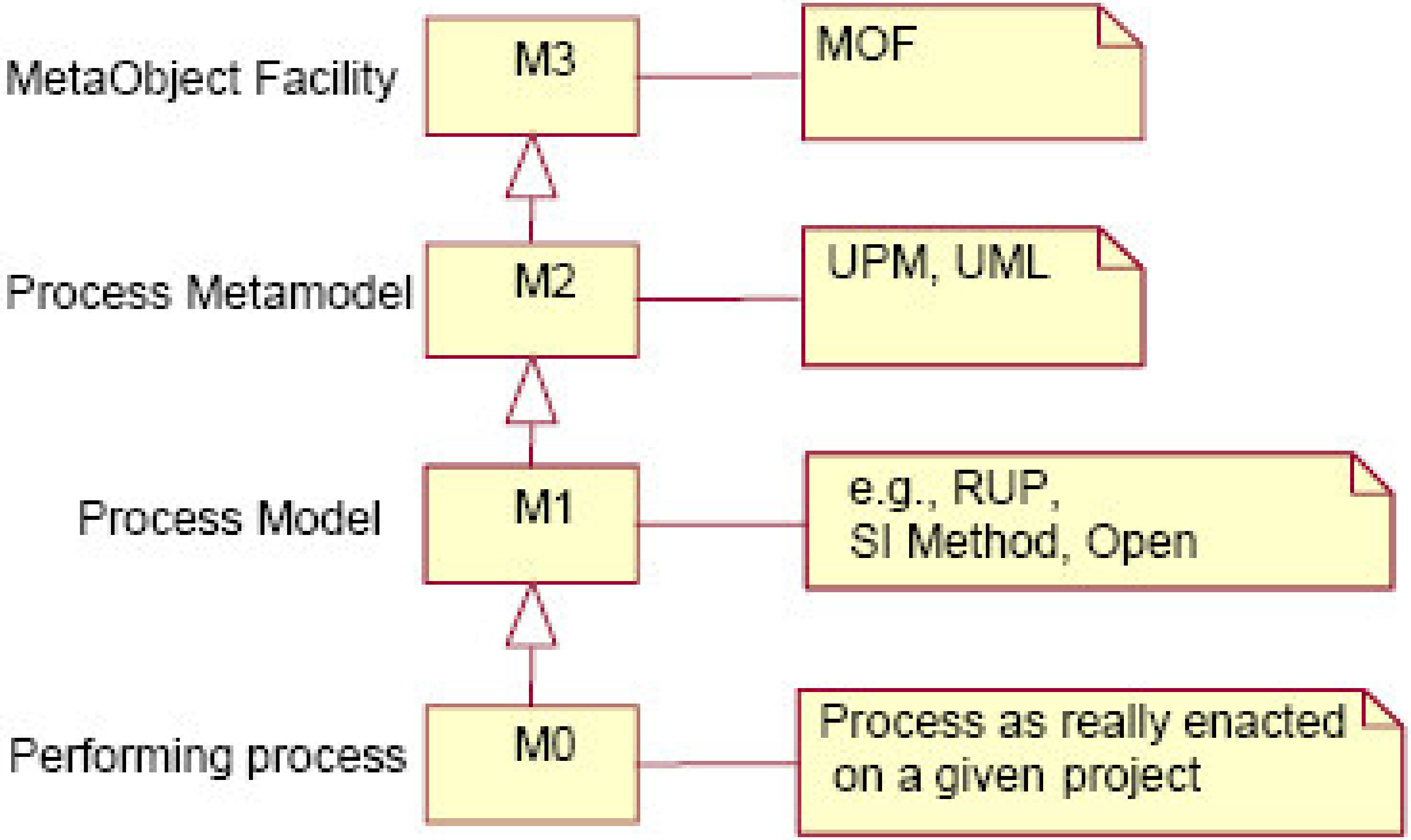


# Process Models

- Model vs. Document
- Best practices and process patterns
- CMMI Level 3:
  - Organization Process Definition
  - Software Process Engineering



# Levels of modeling



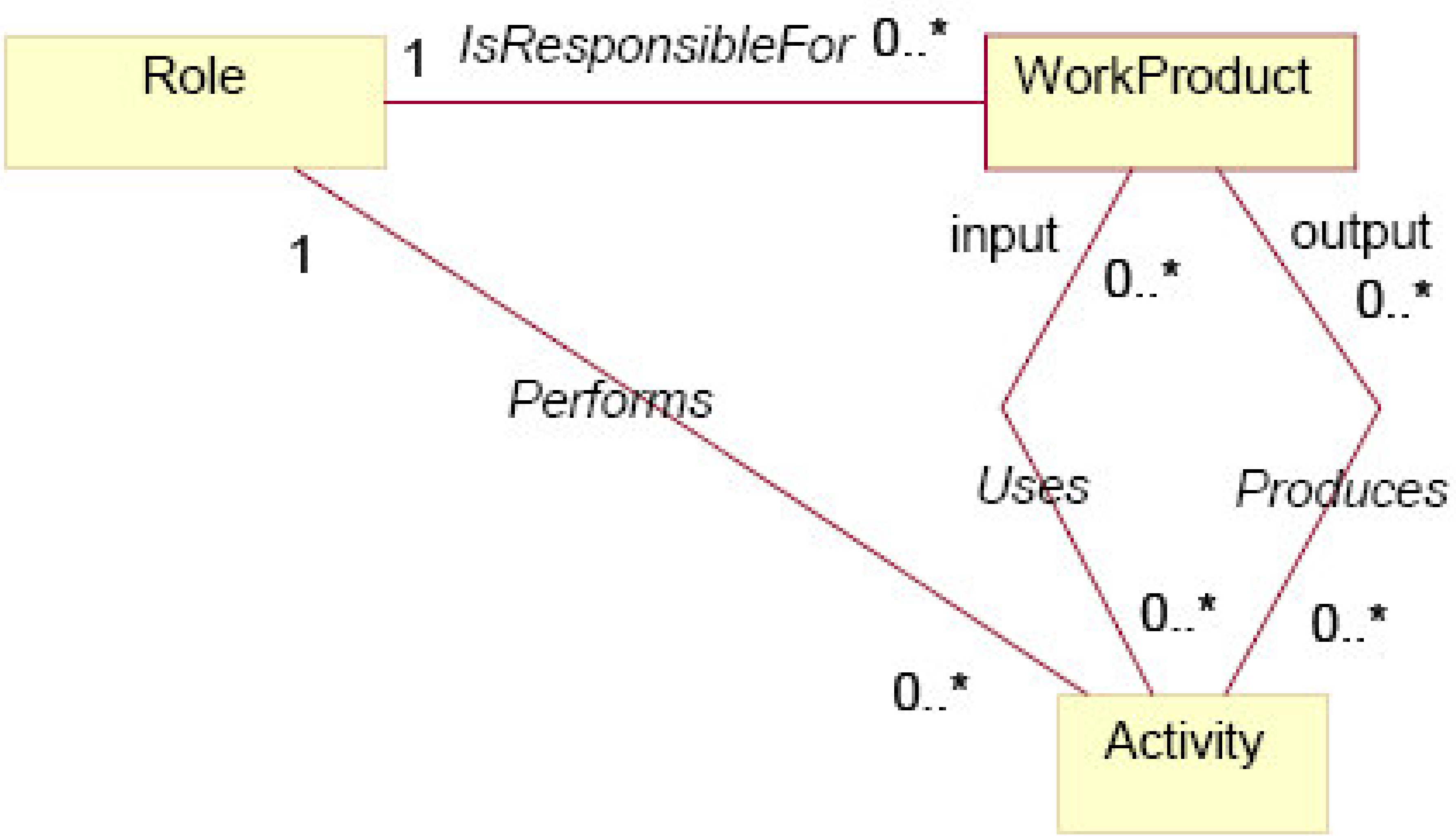
MetaObject Facility — M3 — MOF

Process Metamodel — M2 — UPM, UML

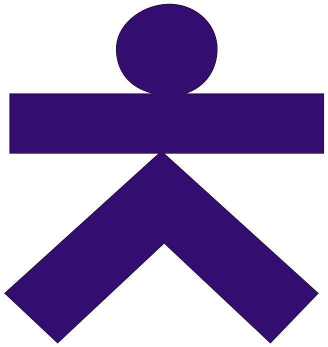
Process Model — M1 — e.g., RUP,
SI Method, Open

Performing process — M0 — Process as really enacted
on a given project

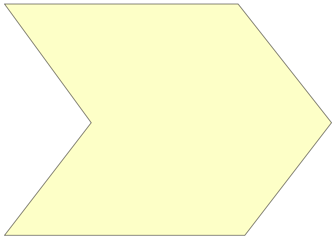
# Conceptual model



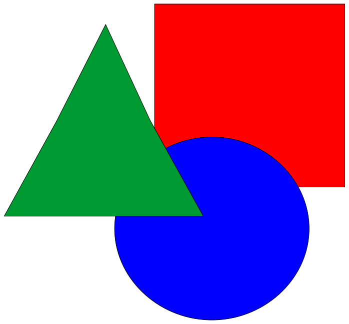
# Graphics notation



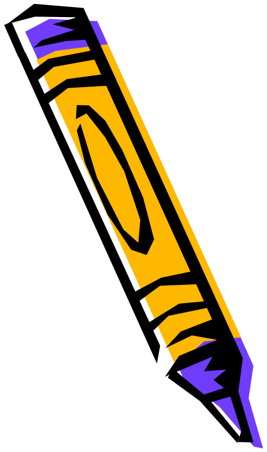
ProcessRole



Activity

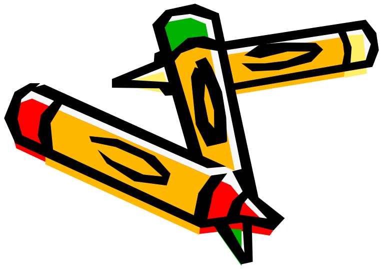


WorkProduct

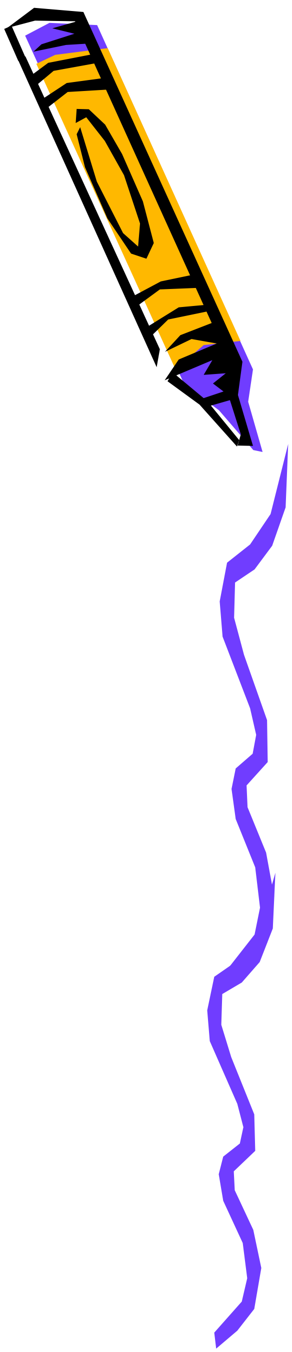


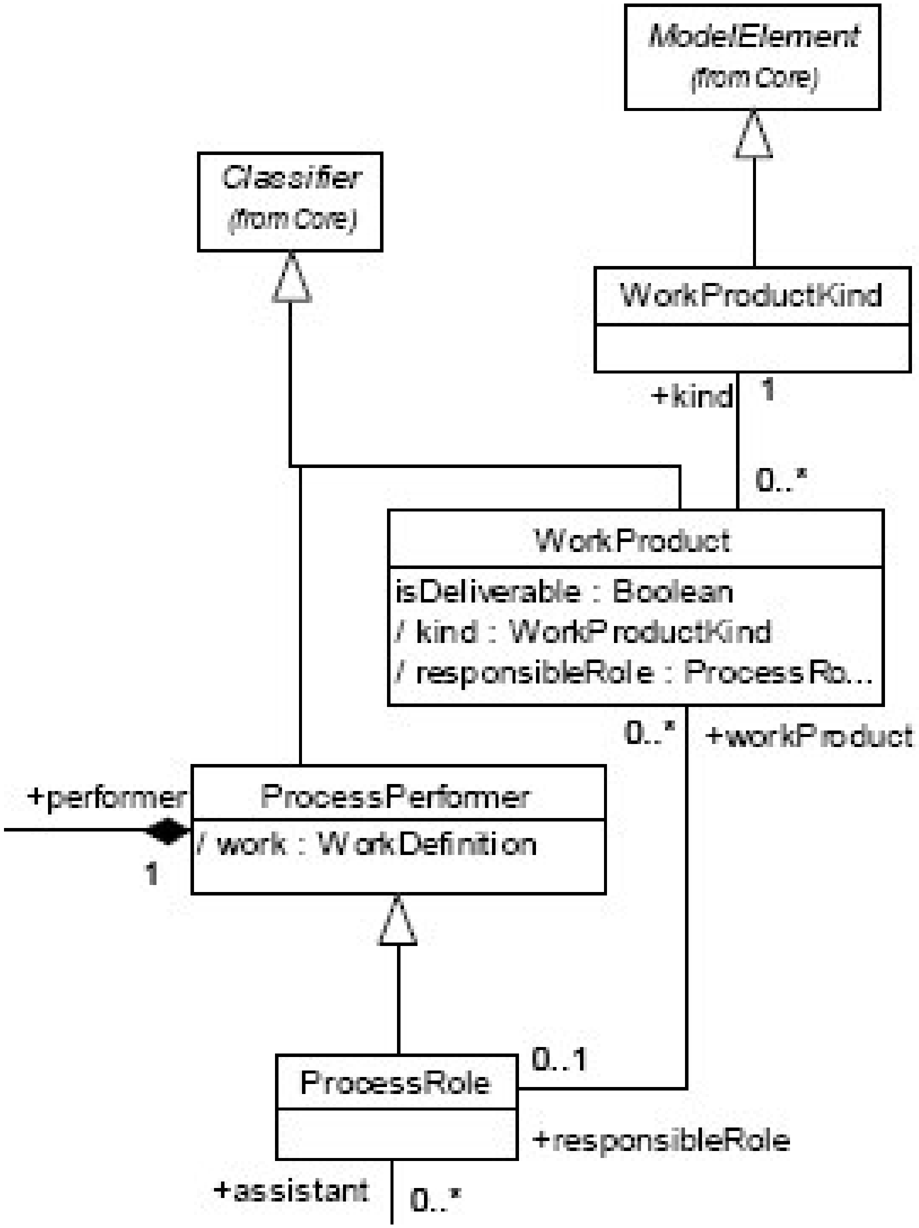
# Main vectors

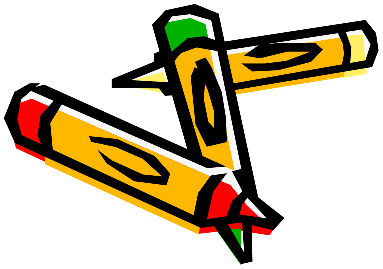
- Team structure
  - ProcessRole, ProcessPerformer
- Workflow
  - Activity, Step, WorkDefinition, …
- Process Artifacts
  - WorkProduct, Document

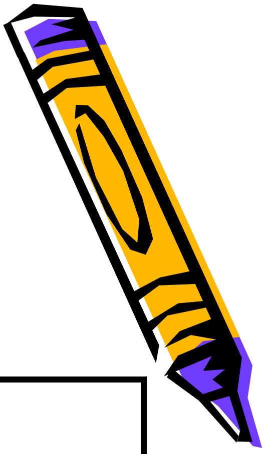


# Team structure

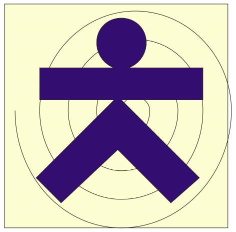
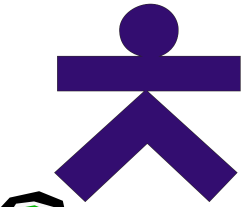


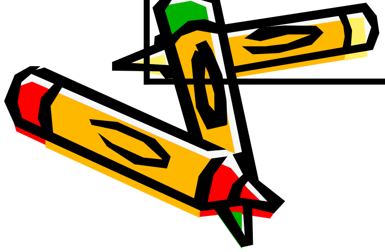




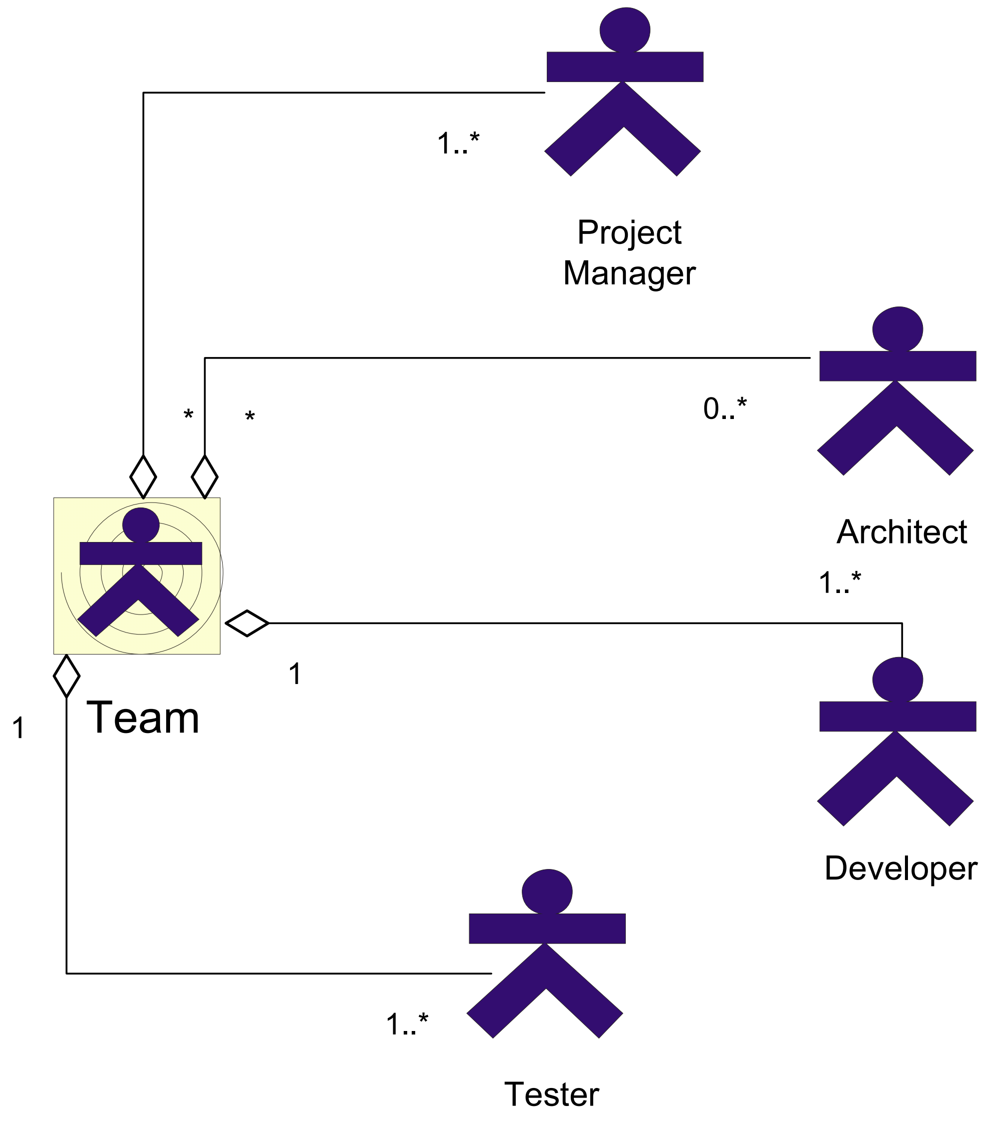


# Team Structure Classes

| | |
|---|---|
| **ProcessPerformer** | |
|  | Describes "process as a whole", is associated with general WorkDefinitions |
| **ProcessRole** | |
|  | Describes concrete roles responsible for specific WorkProducts |



# Team Structure



1..*

Project Manager

*

*

0..*

Architect

1..*

1

Team

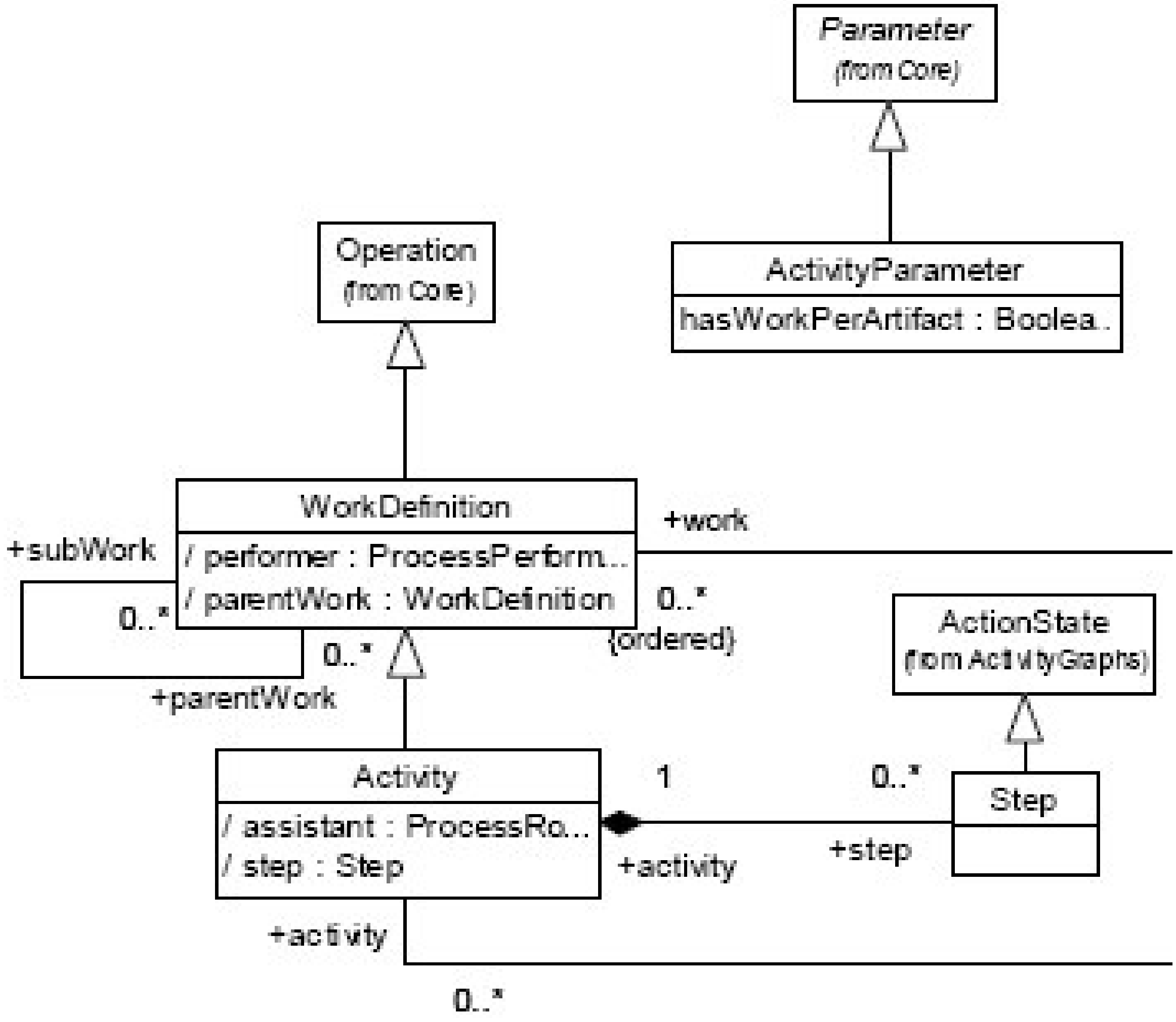
1

Developer

1..*

Tester

# Process workflow



*Parameter*
*(from Core)*

Operation
(from Core)

ActivityParameter
hasWorkPerArtifact : Boolea...

WorkDefinition
/ performer : ProcessPerform...
/ parentWork : WorkDefinition

+subWork
0..*
0..*
+parentWork

+work
0..*
{ordered}

ActionState
(from ActivityGraphs)

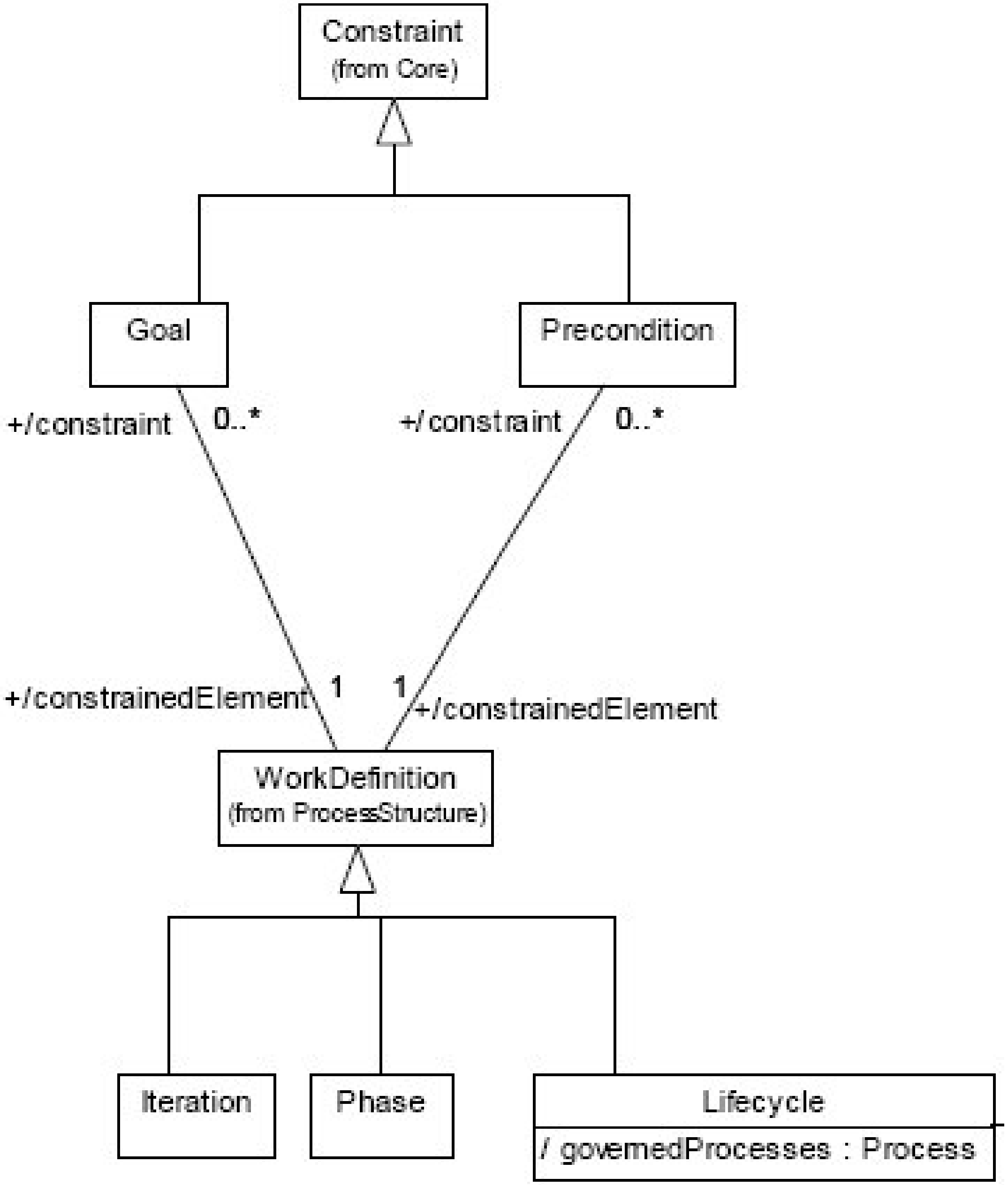
Activity
/ assistant : ProcessRo...
/ step : Step

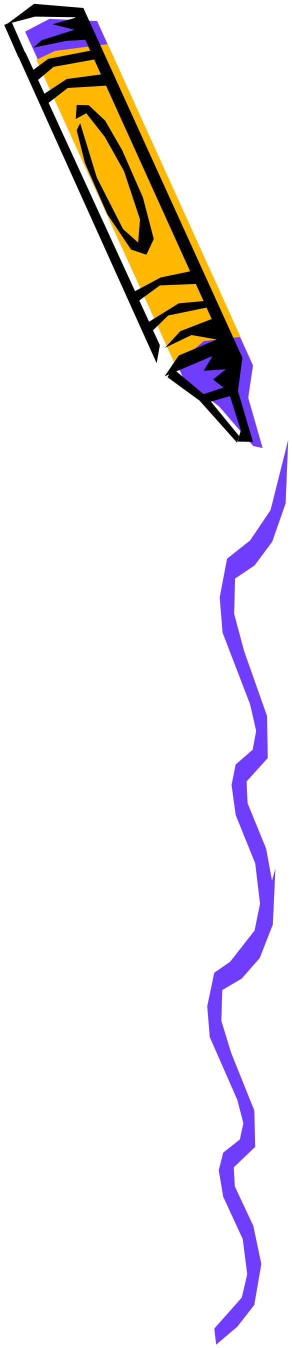
1
+activity
0..*
+step

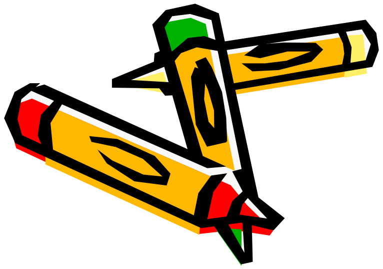
Step

+activity
0..*

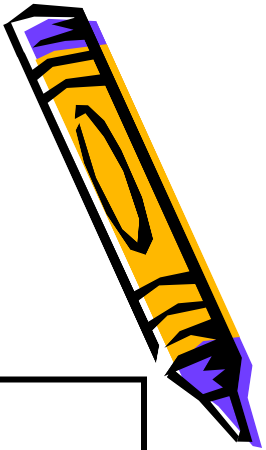
# Process Lifecycle



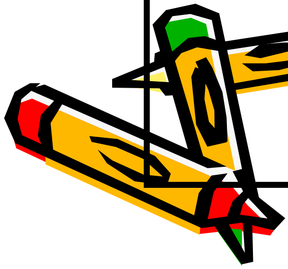




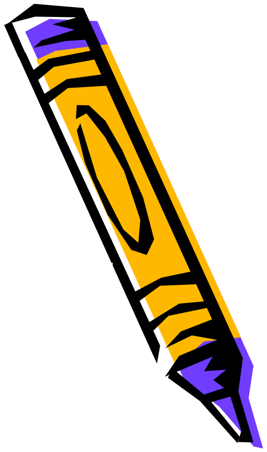
# Work Breakdown Structure

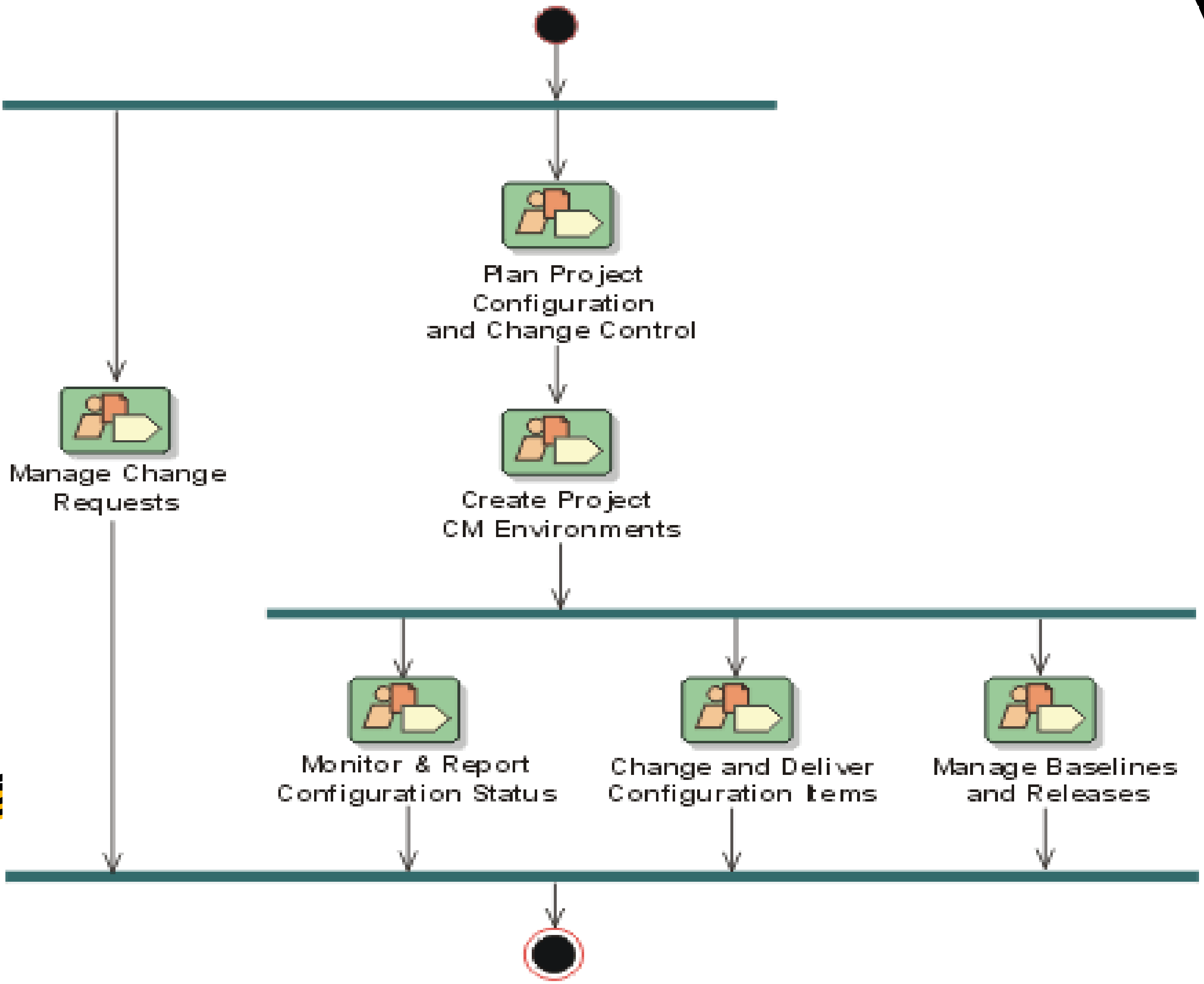


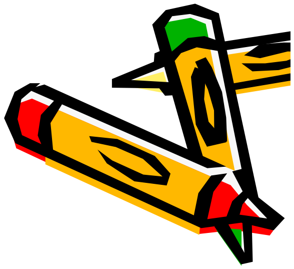
| | | |
|---|---|---|
| | Lifecycle | Sequence of phases to achieve specific goal |
| ❯❯❯ | Phase | Part of process with entry/exit criteria |
| | Iteration | Composite WorkDefinition with minor milestone |
| ❯ | Activity | Piece of work performed by one ProcessRole |
| | Step | Atomic element of Activity |

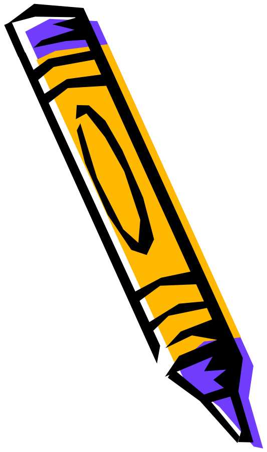


# Configuration Management in RUP



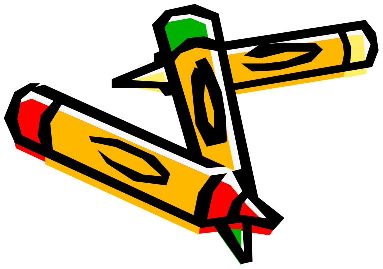

Plan Project Configuration and Change Control
Manage Change Requests
Create Project CM Environments
Monitor & Report Configuration Status
Change and Deliver Configuration Items
Manage Baselines and Releases



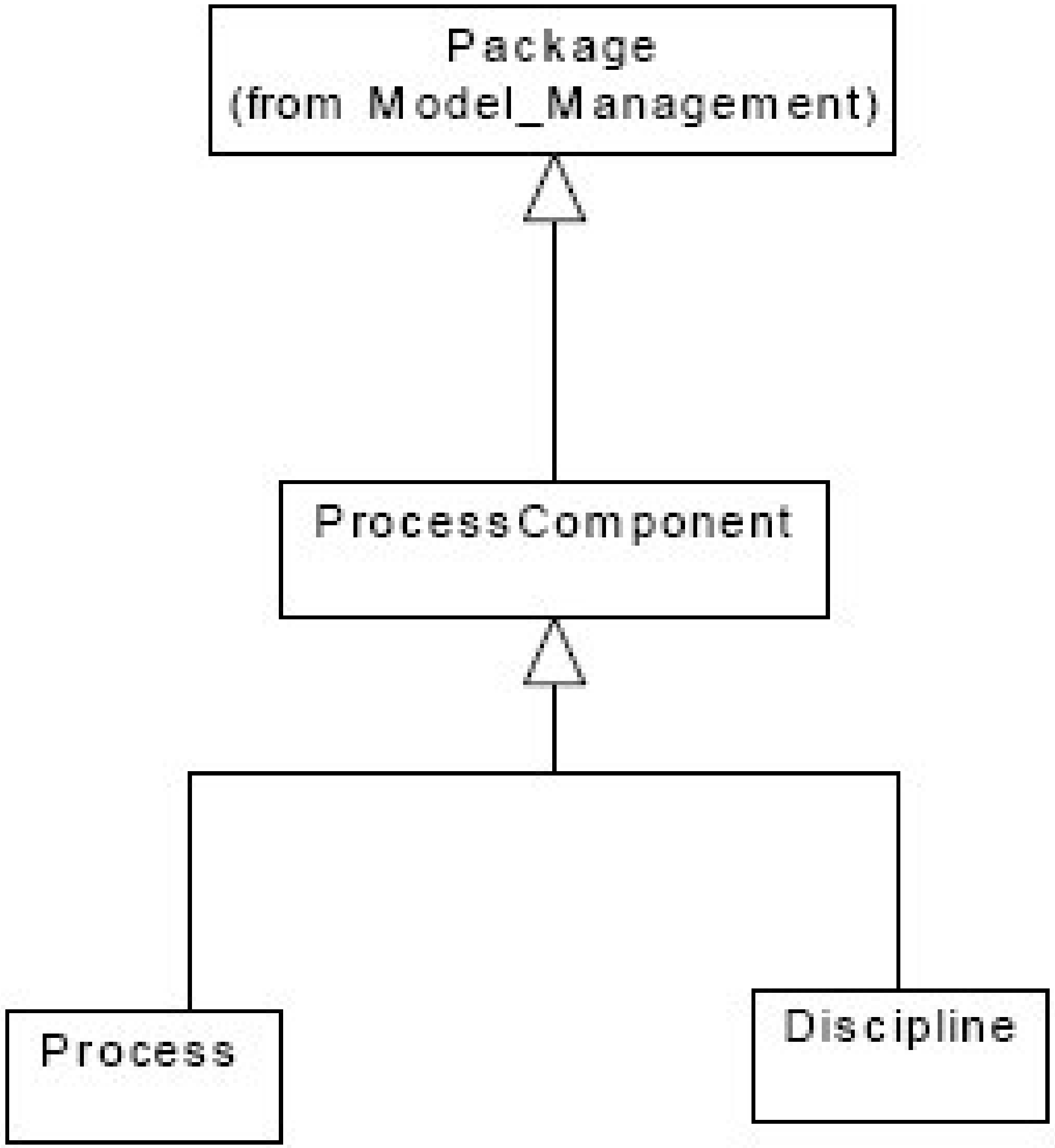


# Constraints

- Precondition and Goal
  - boolean expressions that express state of WorkProducts that are input and output parameters of WorkDefinition
- WorkDefinitions that can be constrained
  - parameters of this WorkDefinition
  - parameters of enclosing WorkDefinition



# Process Components

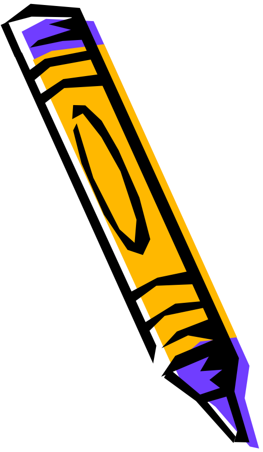


Package
(from Model_Management)

ProcessComponent

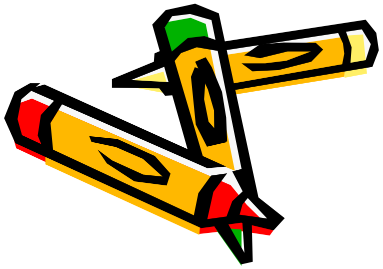
Process

Discipline

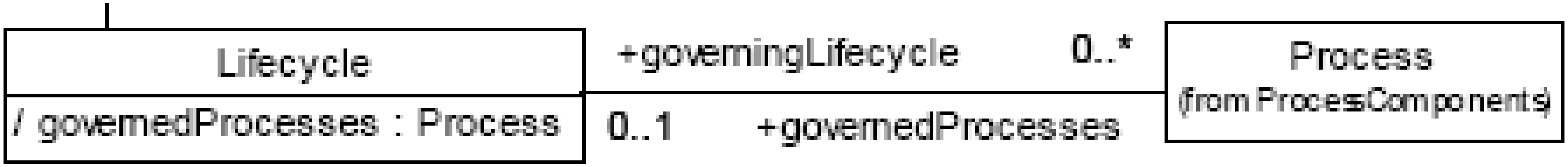


# Model Management

- No semantic meaning, used to structure models
- ProcessComponent
  - Consistent part of process description that can be reused
- Process
  - Complete process that is not to be used as sub-process
- Discipline
  - Used to categorize Activities by “theme”



# They are related…



Lifecycle

/ govemedProcesses : Process

+govemingLifecycle

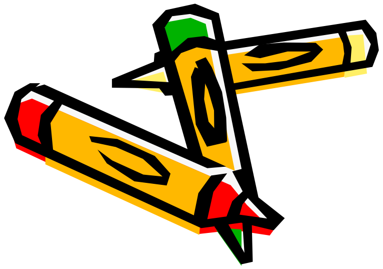
0..*

0..1

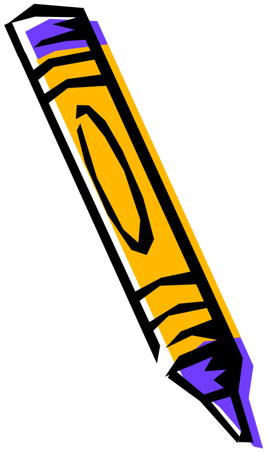
+govemedProcesses

Process

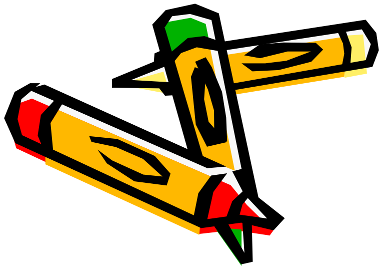
(from ProcessComponents)



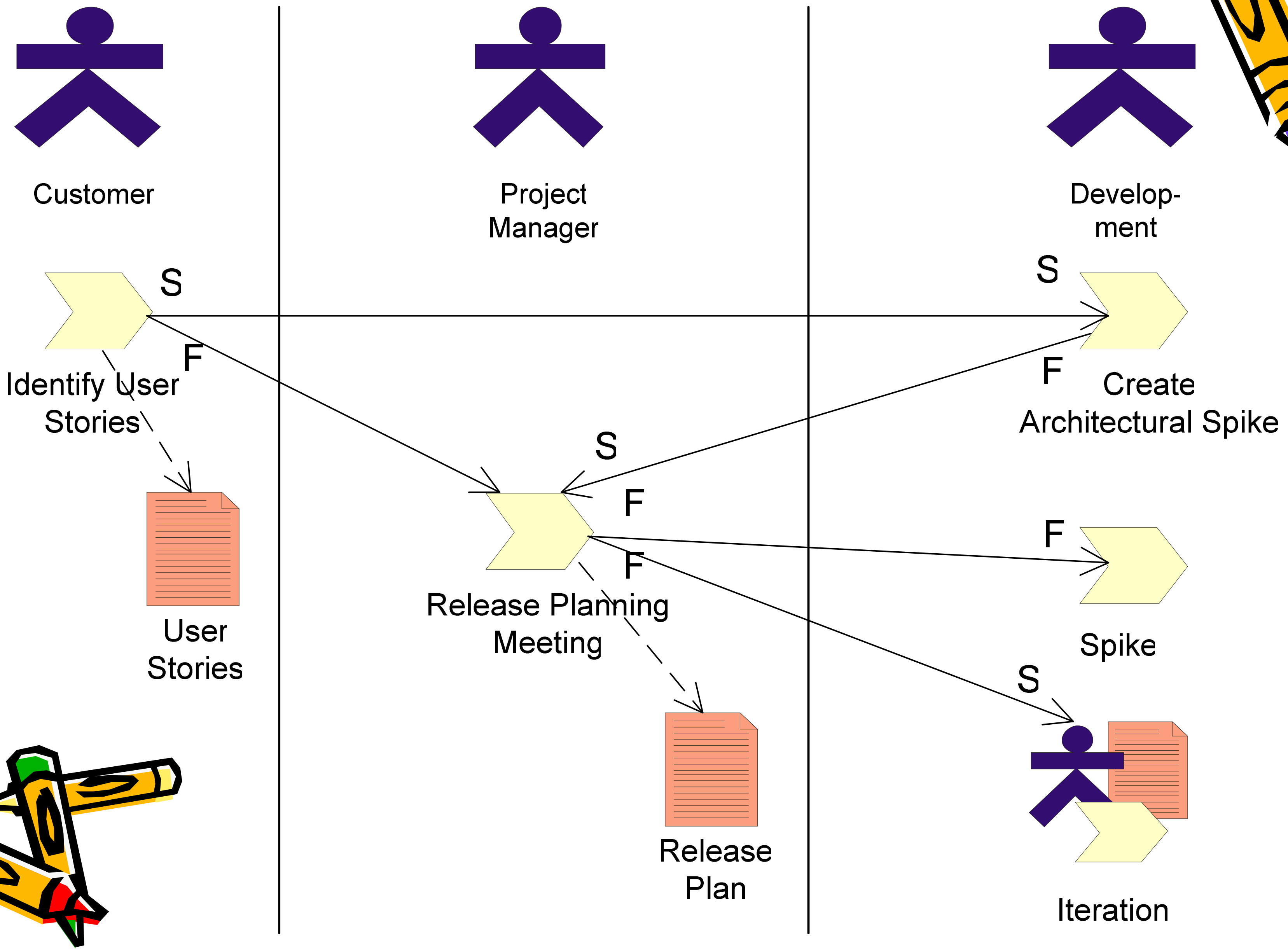
# Process and Lifecycle



- Lifecycle
  - describes behavior of the process
- Process
  - contains all descriptive material in the process
- One Lifecycle can drive several processes



# Example from XP



Customer

Project Manager

Develop-ment

S

Identify User Stories

F

User Stories

S

Create Architectural Spike

F

S

F

Release Planning Meeting

F

F

Spike

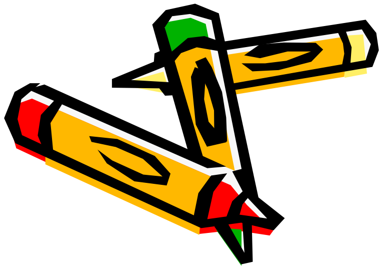
Release Plan

S

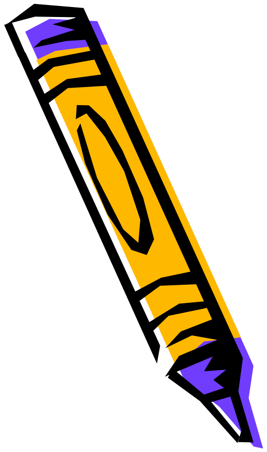
Iteration

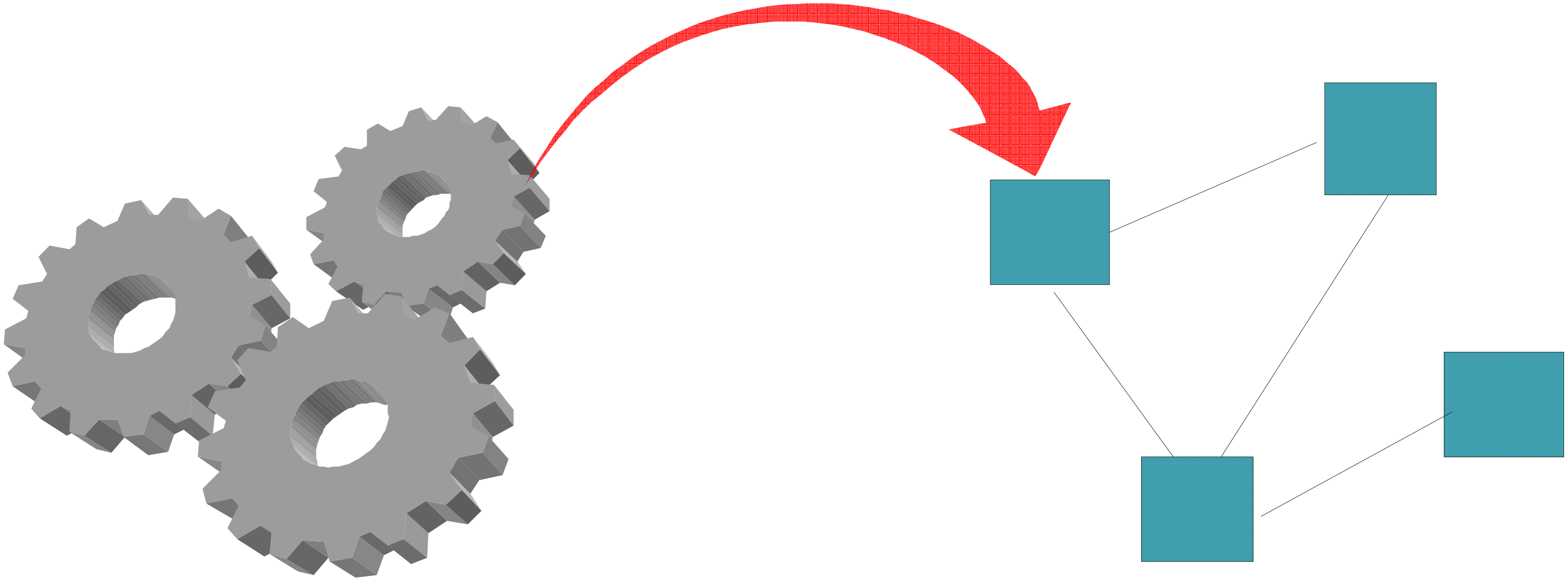
# SPEM Essentials

- Can define roles, responsibilities and artifacts
  - via associations of Process Roles with Activities and WorkProducts
- Can define WBS
  - via composition of WorkDefinition descendants (LifeCycle, Phase, Iteration)

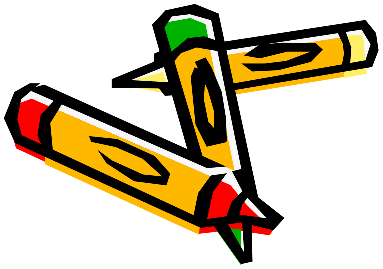


# SPEM in Software Engineering

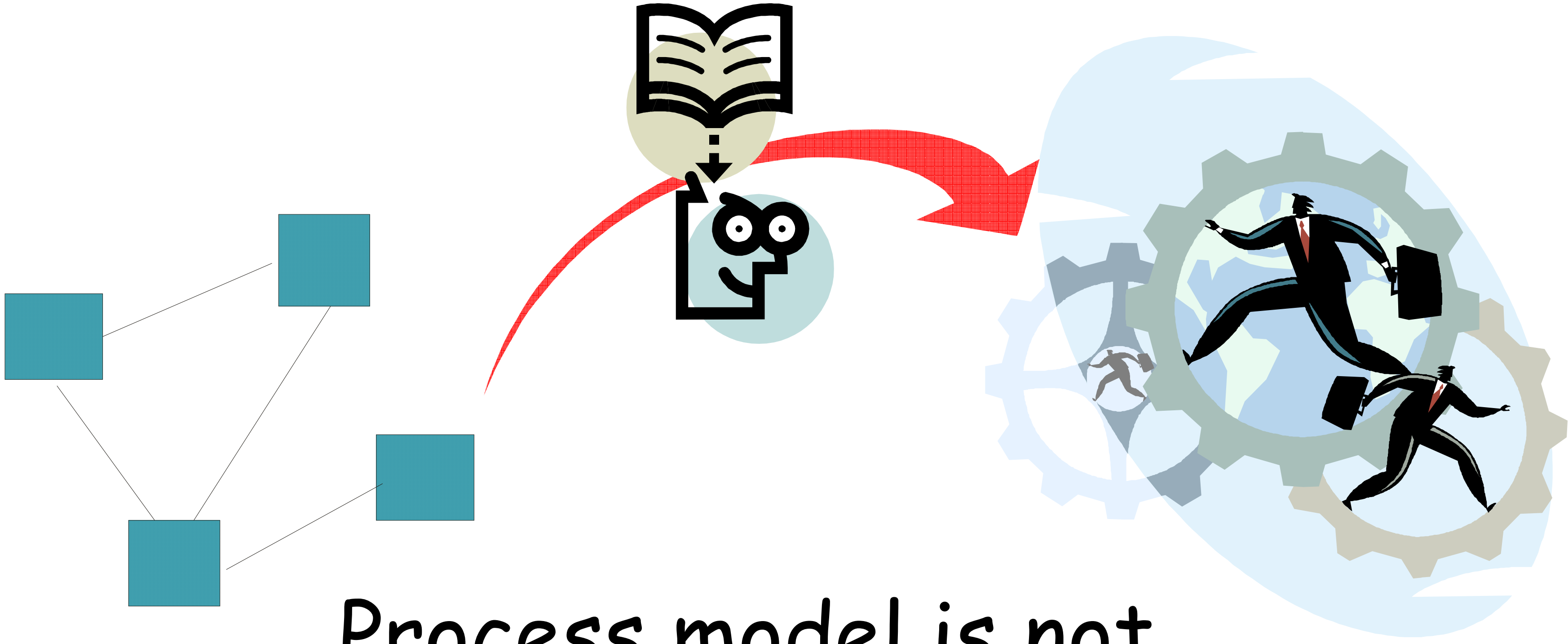




- Share knowledge
- Capture best practices

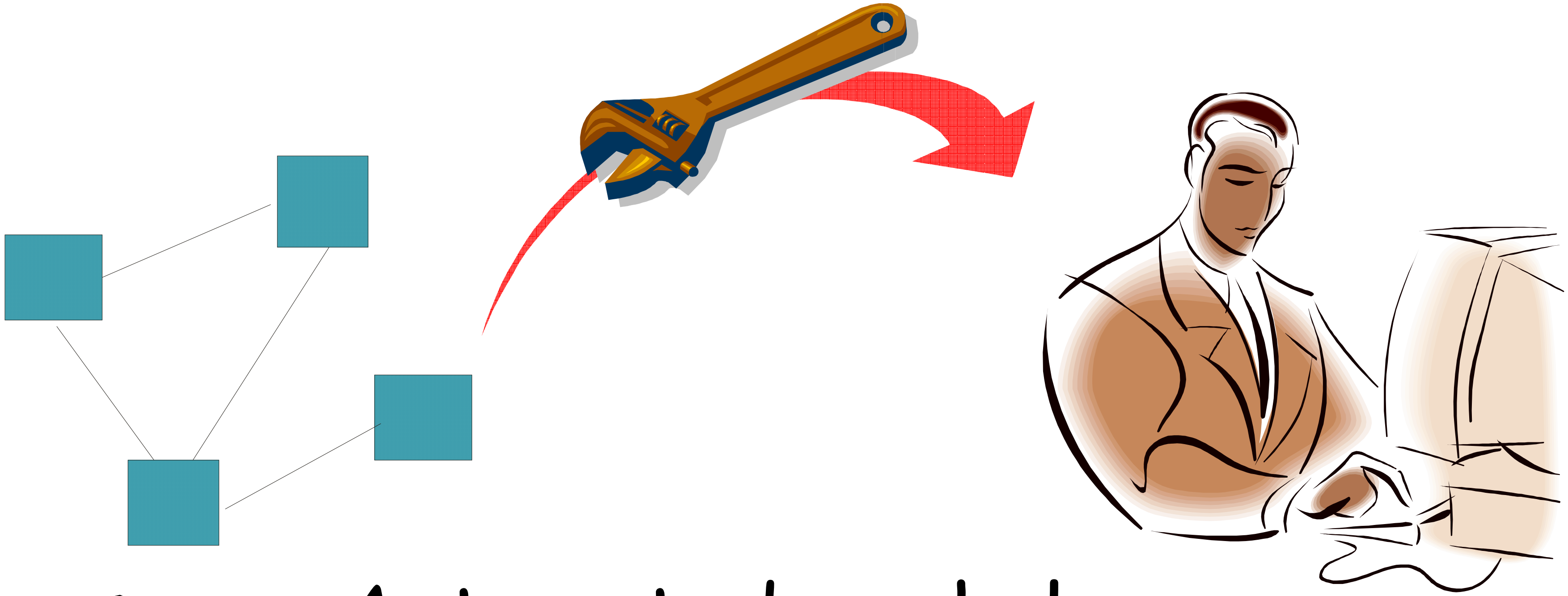


# Process Engineering



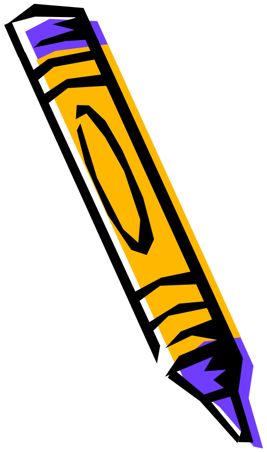
Process model is not first class citizen in PE

# Process Engineering

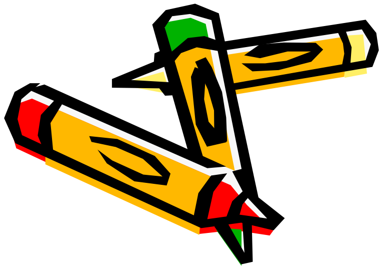


Automated workplace

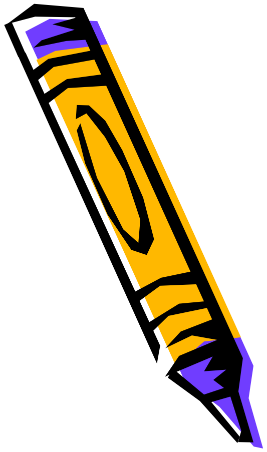
# Automated Process Engineering



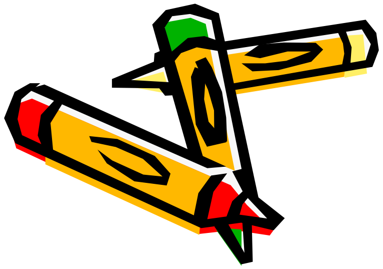
- Rational Unified Process
  - Rational Process Workbench
- Microsoft Visual Studio Team System
  - Process templates
  - Can be used to enact *any* process
  - IRIS Process Author (http://www.osellus.com/scenarios/iris_vs2005.html )



# 5 Things Process Engineer Should Know

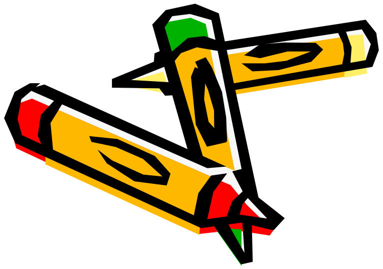


- More about structure as a whole than about relations
- Start with concrete WorkProducts (deliverables) and ProcessRoles
- Use “middle” level of abstraction when modeling workflow for the first time
- Do not escape from SPEM using Guidance
- Actual process is an instance of the model

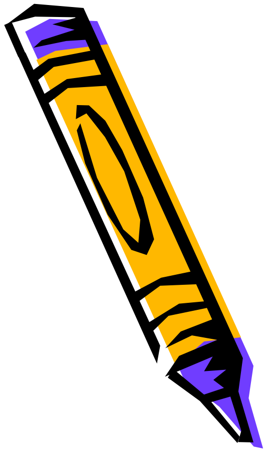


# More information

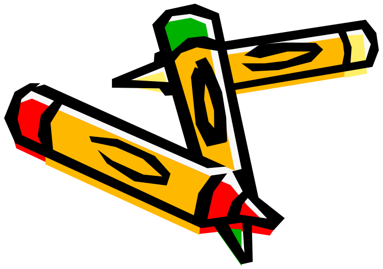
- OMG Unified Modeling Language Specification v 1.5
  - http://www.omg.org/technology/documents/formal/uml.htm
- Software Process Engineering Metamodel Specification v 1.0
  - http://www.omg.org/technology/documents/formal/spem.htm



# Tools that support SPEM



- Enterprise Architect
  - http://www.sparxsystems.com.au
- Objecteering/UML
  - http://www.objecteering.com



# Quiz

- SPEM can be used to model only RUP-based processes.
- SPEM allows to define WBS for processes.
- SPEM can be used to define schedule for future processes.
- What tools support SPEM?