

ZIP! Becoming Seamless

An Efficient Integrated Process from Stakeholder Trigger to Component Architecture

Tutorial at

Object Technology 2000

Oxford, UK, March 2000

Graham McLeod

mcleod@iafrica.com

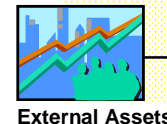


www.inspired.org

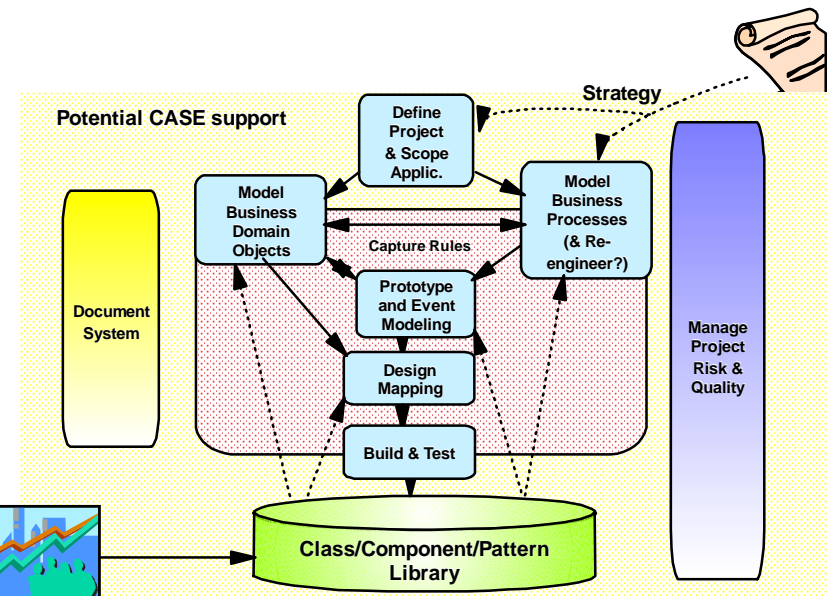


University of Cape Town

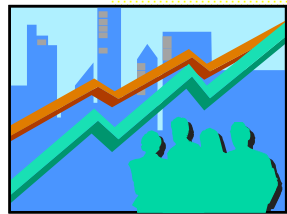
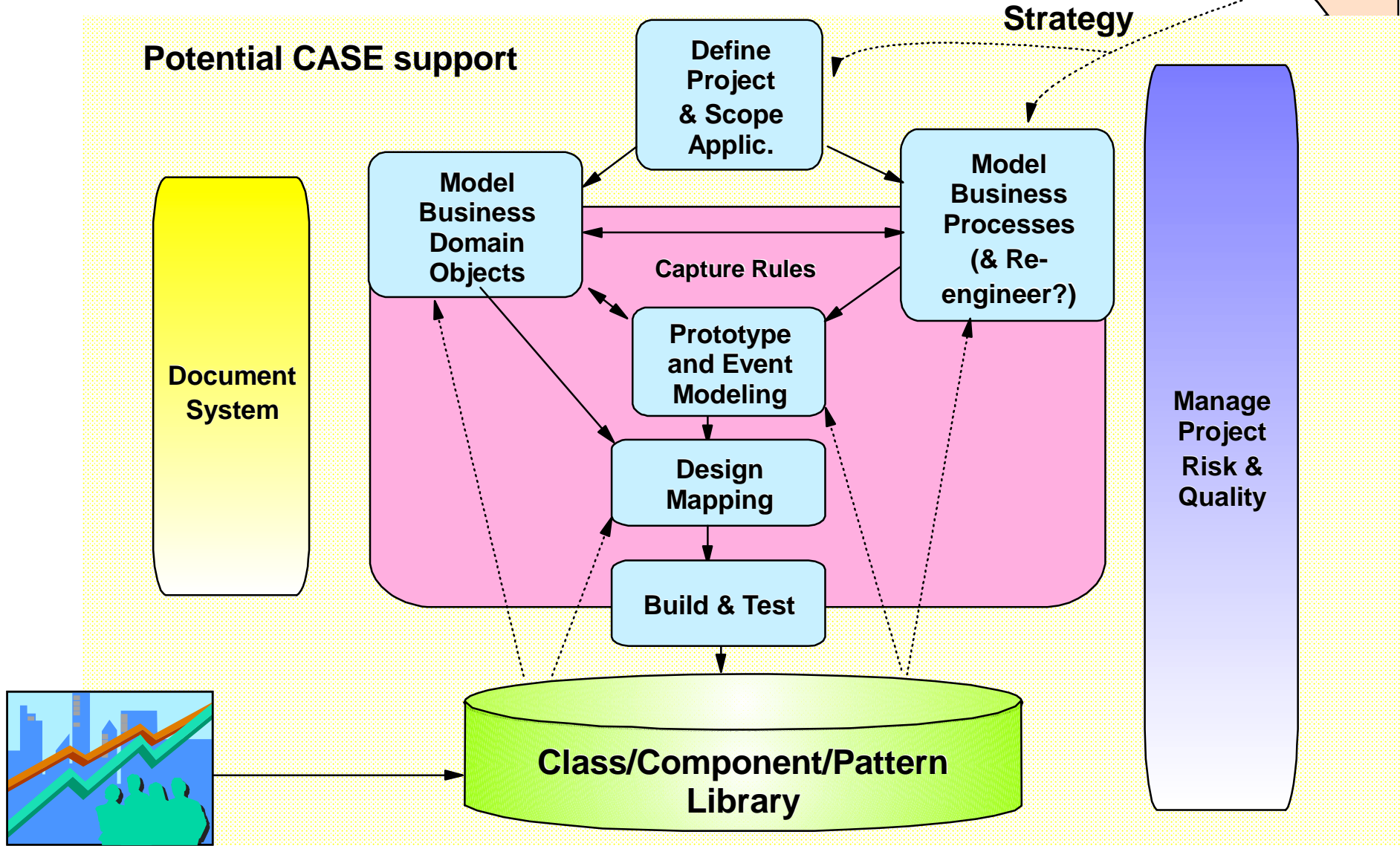
www.uct.ac.za



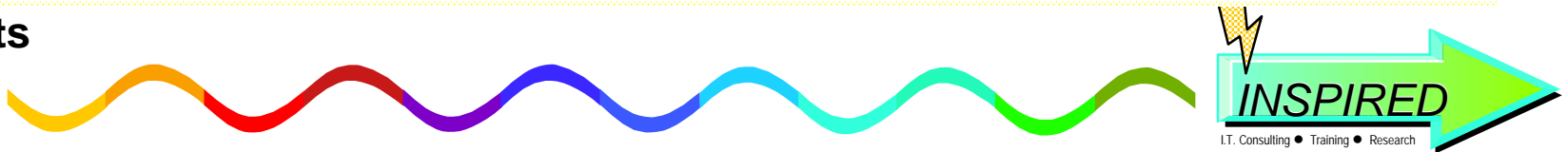
External Assets



Inspired Method Overview

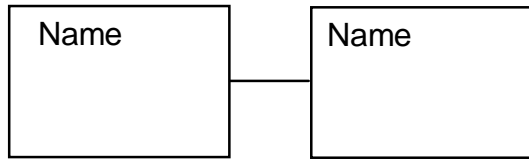


External Assets

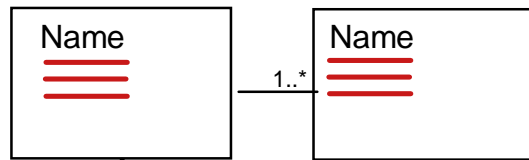
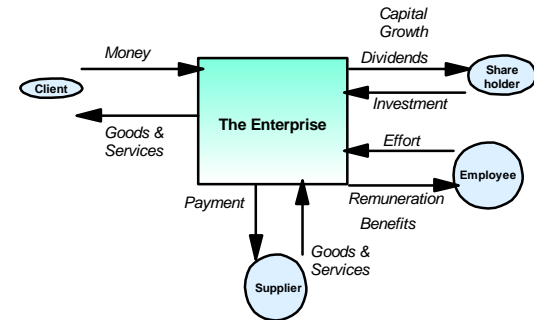


I.T. Consulting • Training • Research

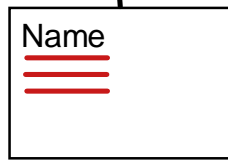
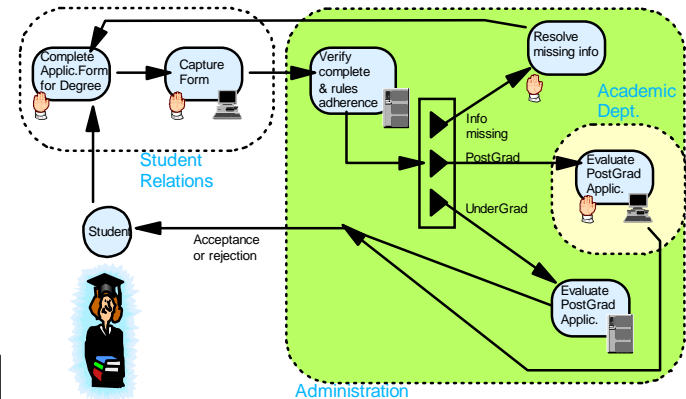
Increasing Fidelity



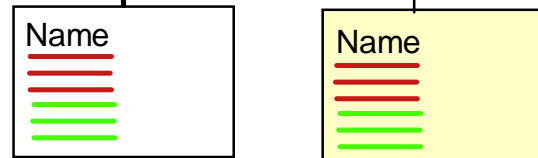
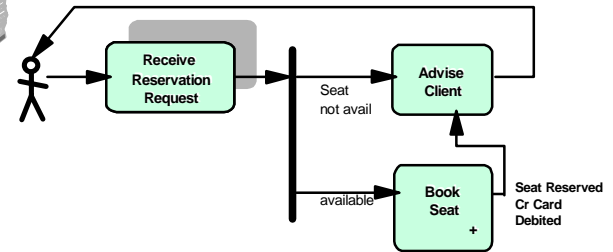
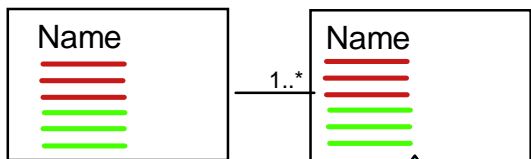
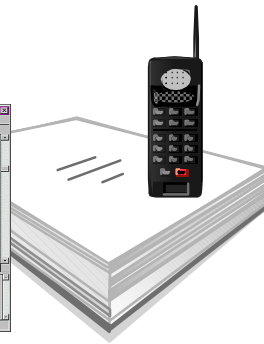
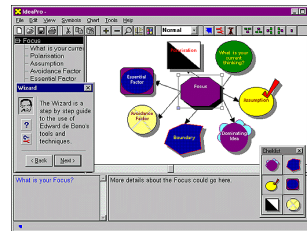
Early Analysis



Late Analysis



Prototyping

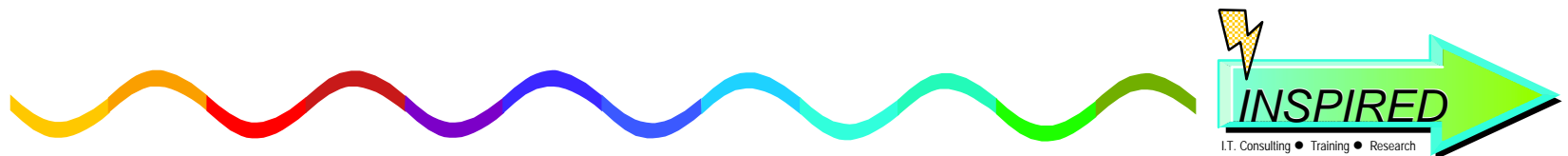
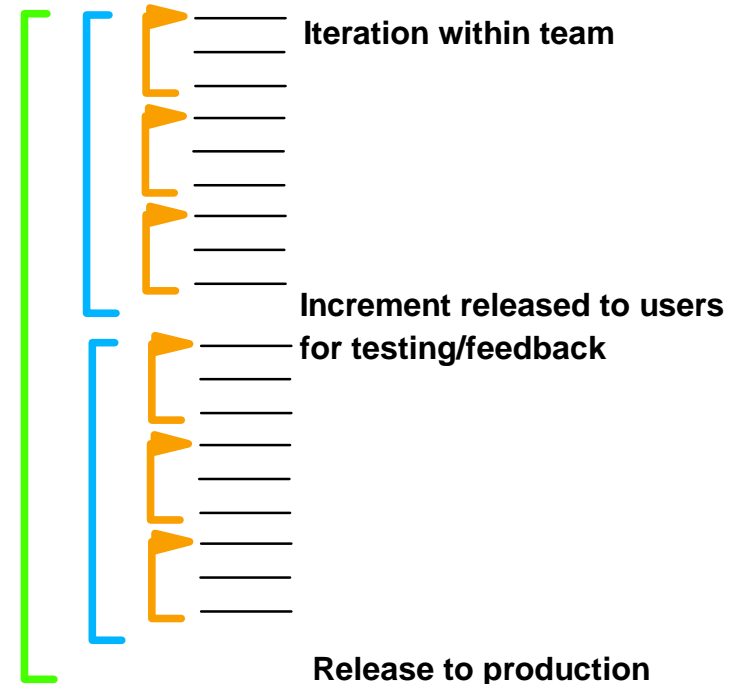


Design

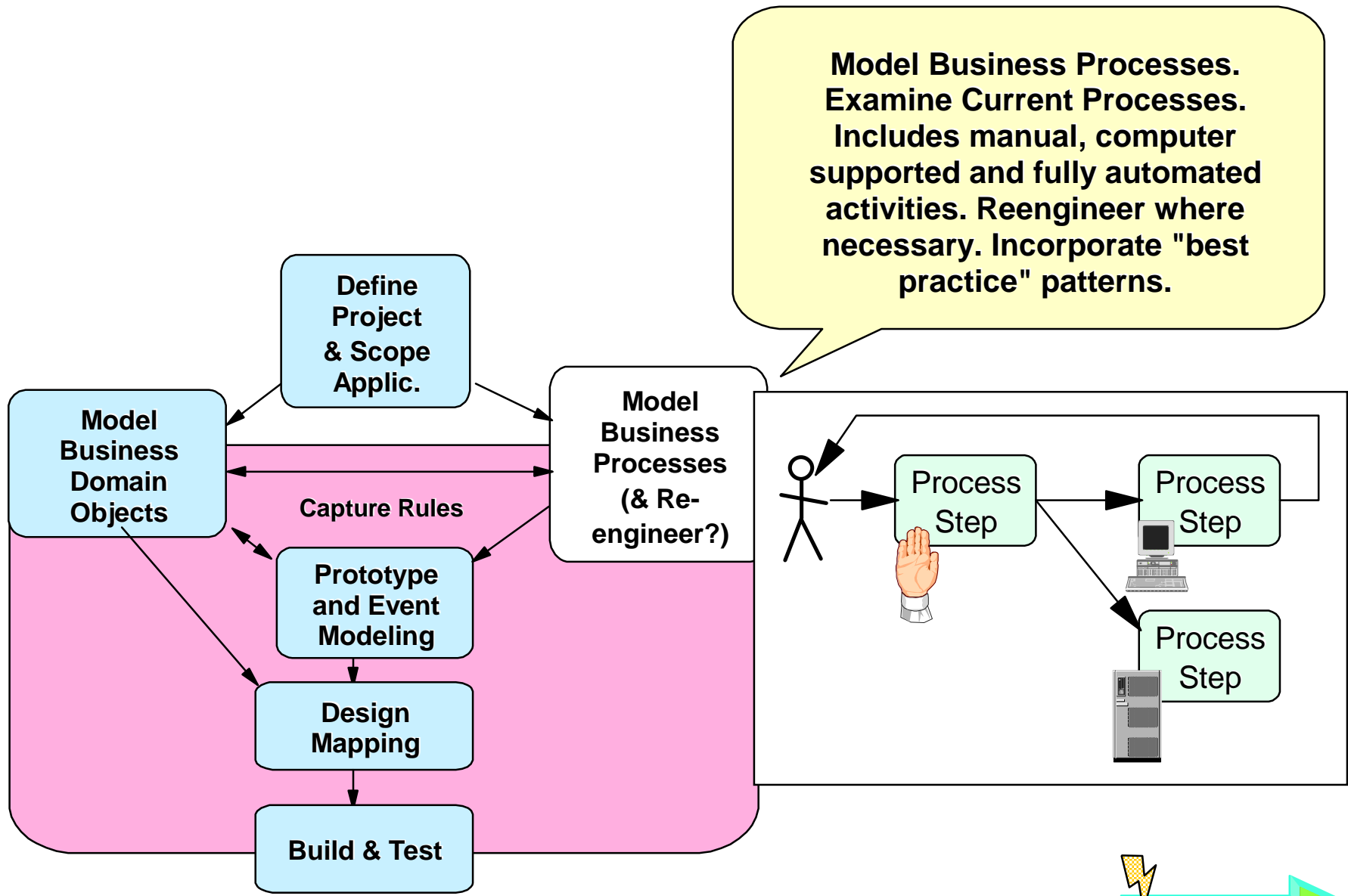


Incremental, Iterative

- Functionality delivered to users in increments (small, frequent releases)
- Iteration through activities to reach desired state of a deliverable
- All deliverables subject to peer review and "testing" when produced
- Routine regression testing
 - Harnesses and stubs
 - Workspaces
 - Test Beds
 - Automated comparators

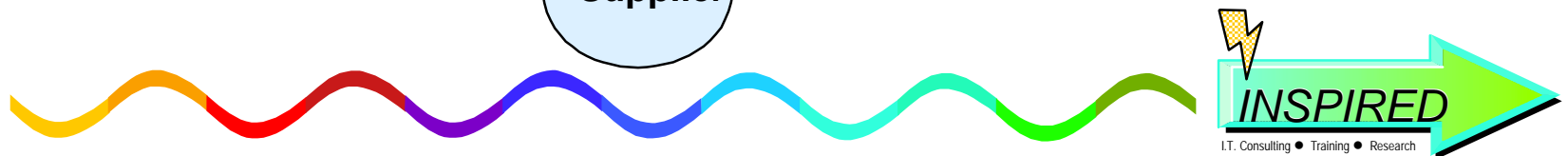
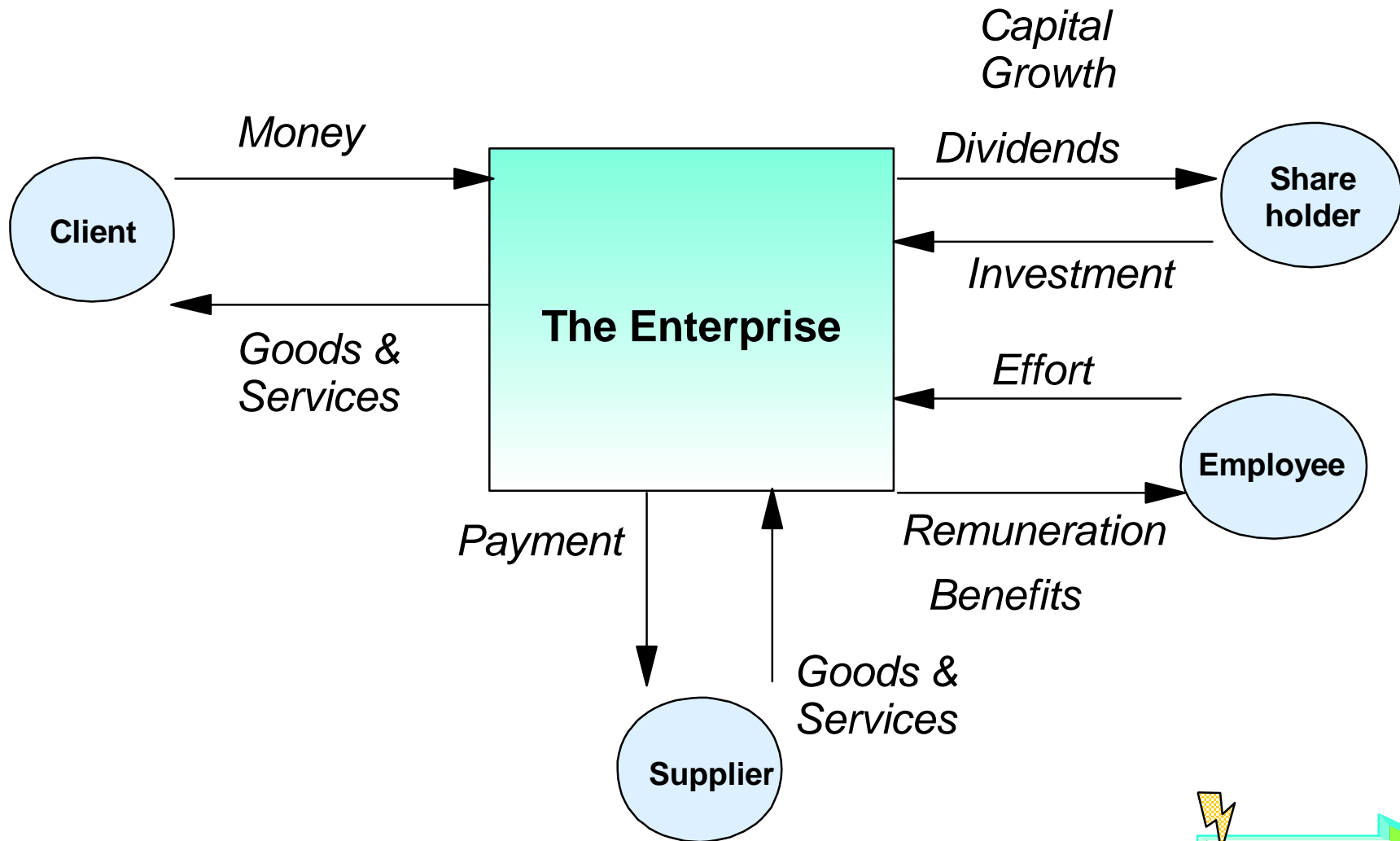


Model Business Processes



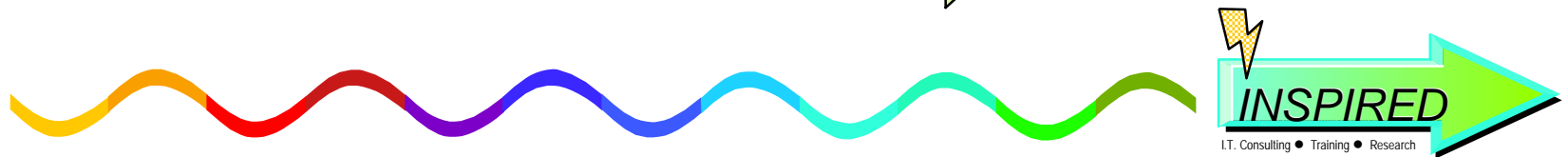
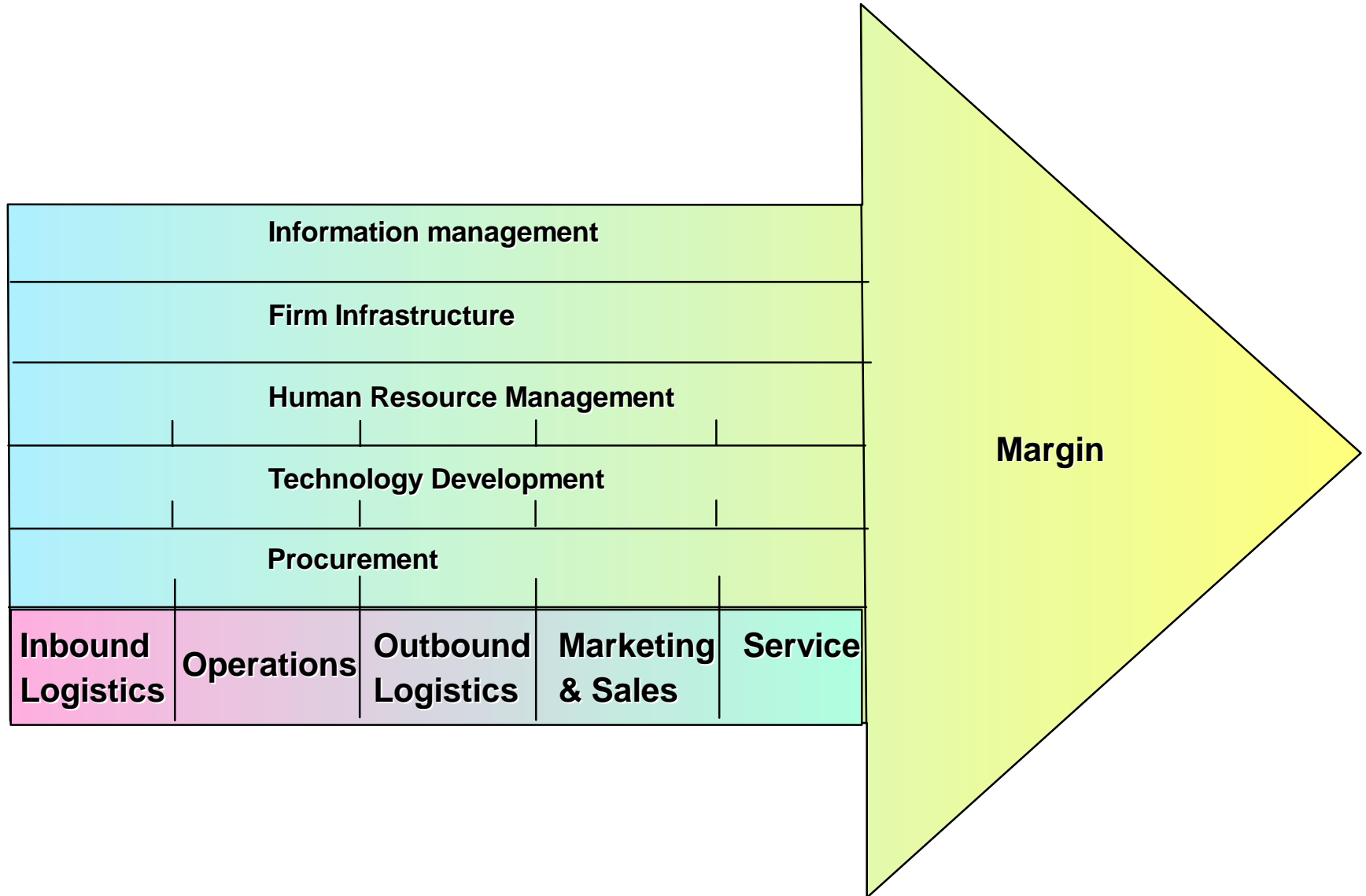
Stakeholder Perspective

- Focus is on the Enterprise Boundary

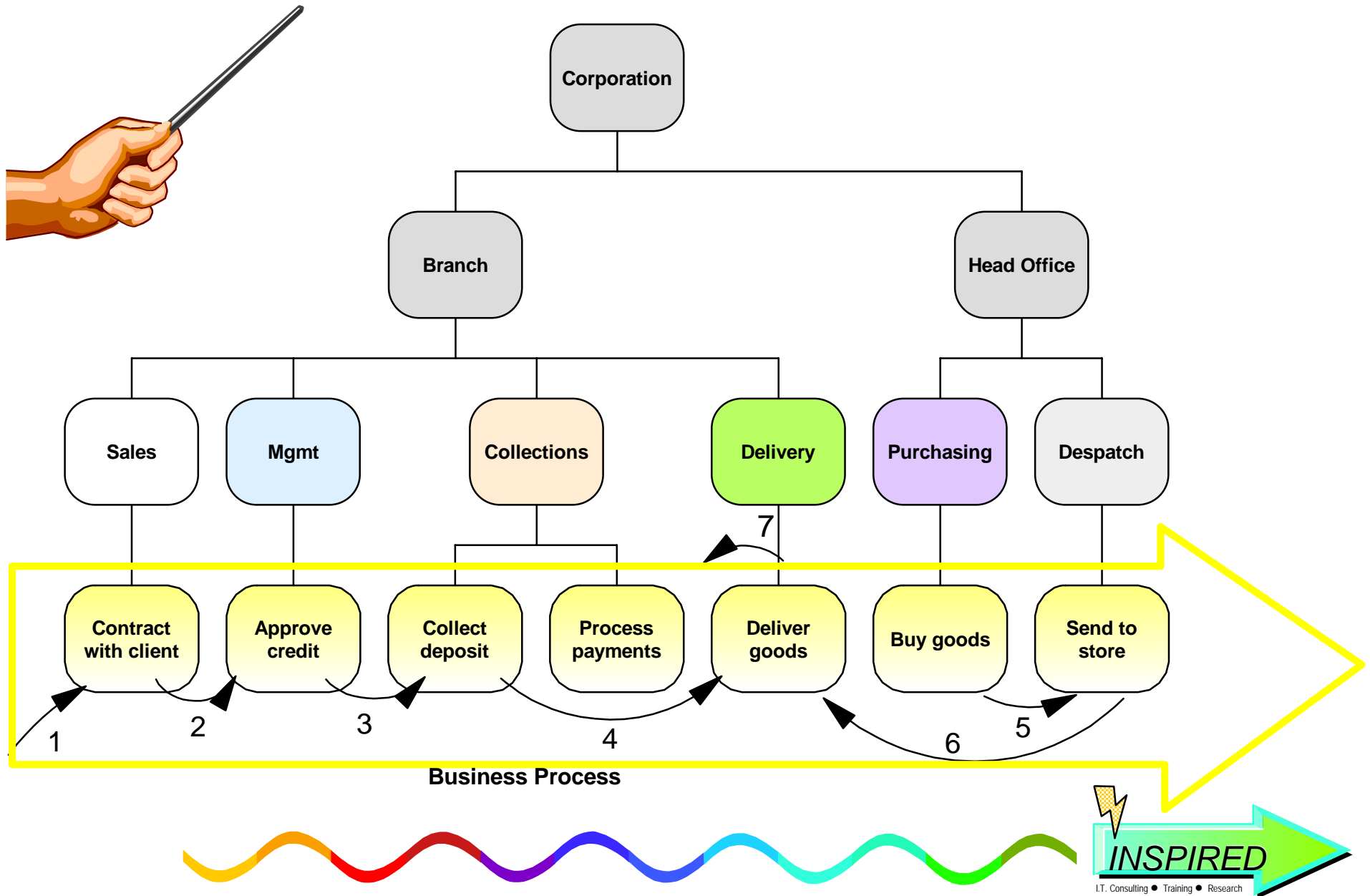


Porter Value Chain model

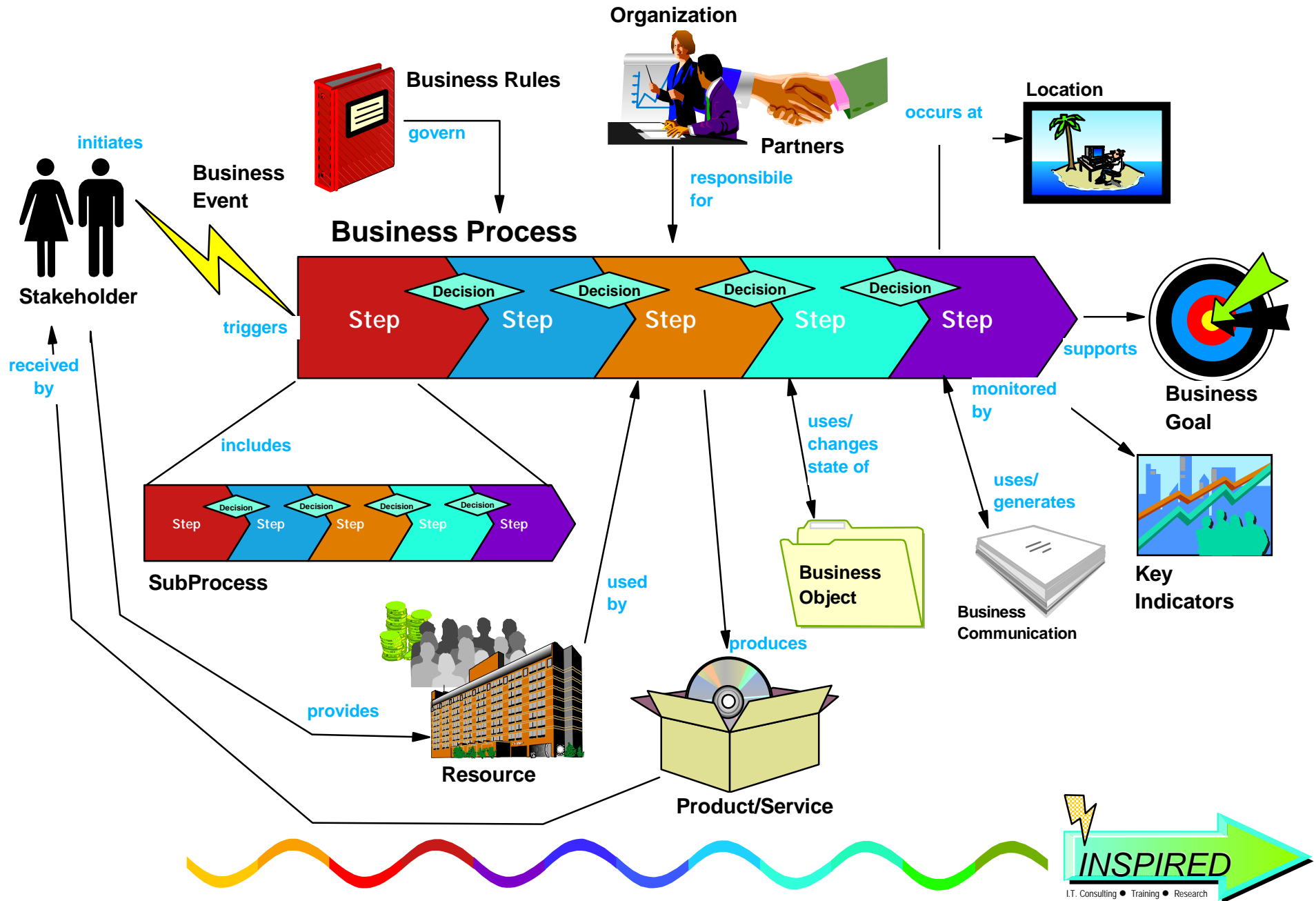
Support Activities



Value Chain cuts across functional boundaries

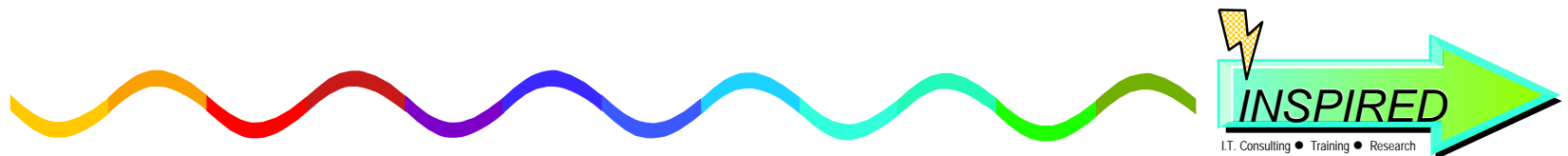


Process Architecture

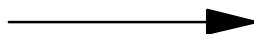


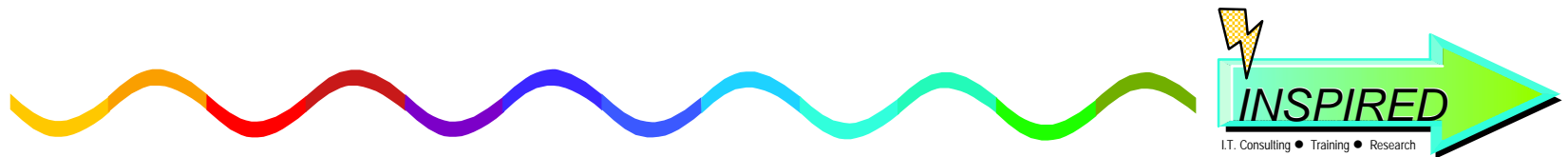
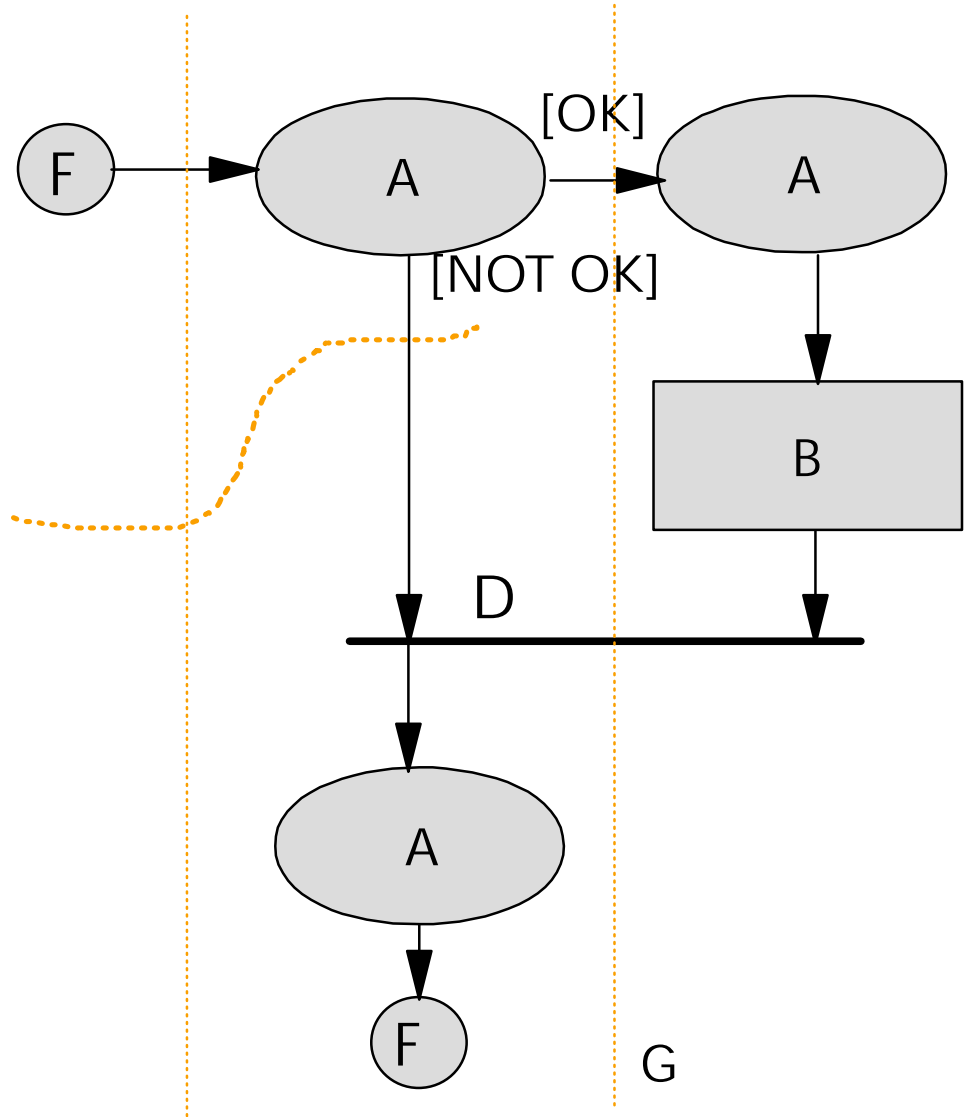
Sources of ideas

- Previous work on integrating Memo and Martin/Odell OOA method
 - Stakeholder models
 - Value chain identification (ala Porter)
 - Business process models (ala GMD)
 - Integration to Martin/Odell event models
- Design mapping to layered architecture based upon MVC (Smalltalk/Xerox PARC)
- Classical BPR principles (Hammer, Champey, Davenport)
- David Taylor (Enterprise Engines) convergent engineering approach
- *Inspired* Architecture and Strategy work
- UML variant in '98
- Process Modeling extensions at Tools '99



Standard UML Notation

- A. Activity
- B. External Activity
- C. Triggers 
- D. Synchronisation Bar
- E. Annotation
 - We use to note outcomes
- F. Start and Terminate
- G. Swim lane



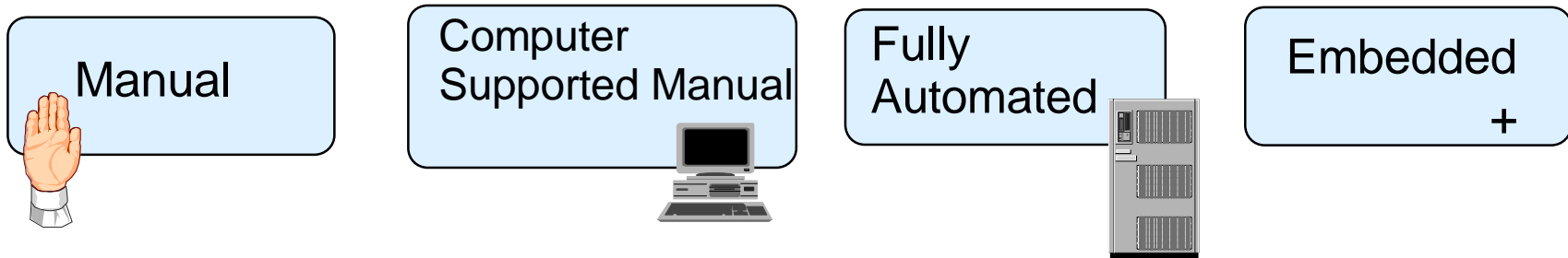
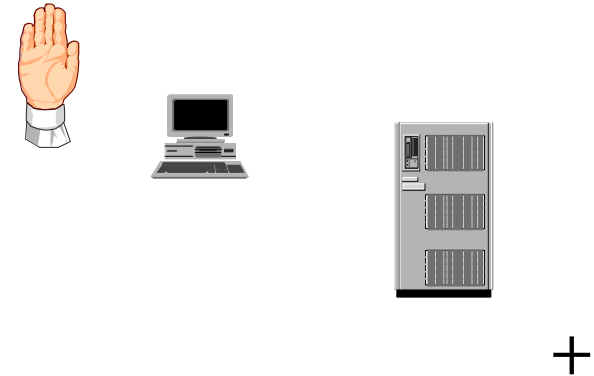
Denoting activity types

- Extend by using UML stereotype mechanism to denote type of activity

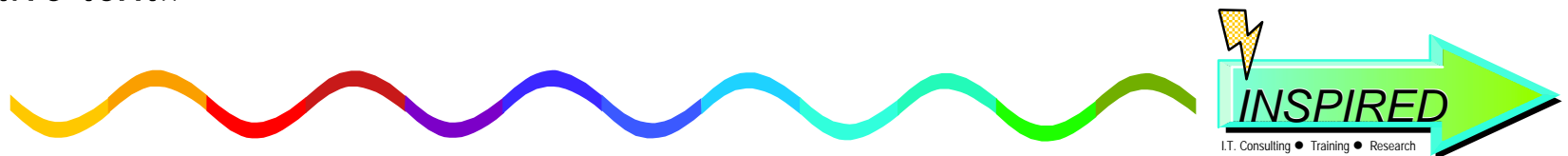
- Text

- <<manual>>
- <<supported>>
- <<automated>>
- <<embedded>>

- Icons



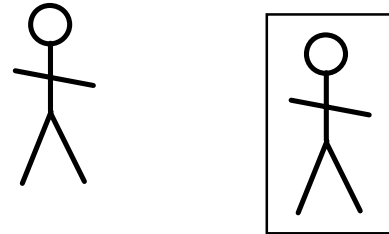
We tend to use rounded rectangles in place of ovals, since it easier to fit descriptive text..



Agents, Inputs, Outputs

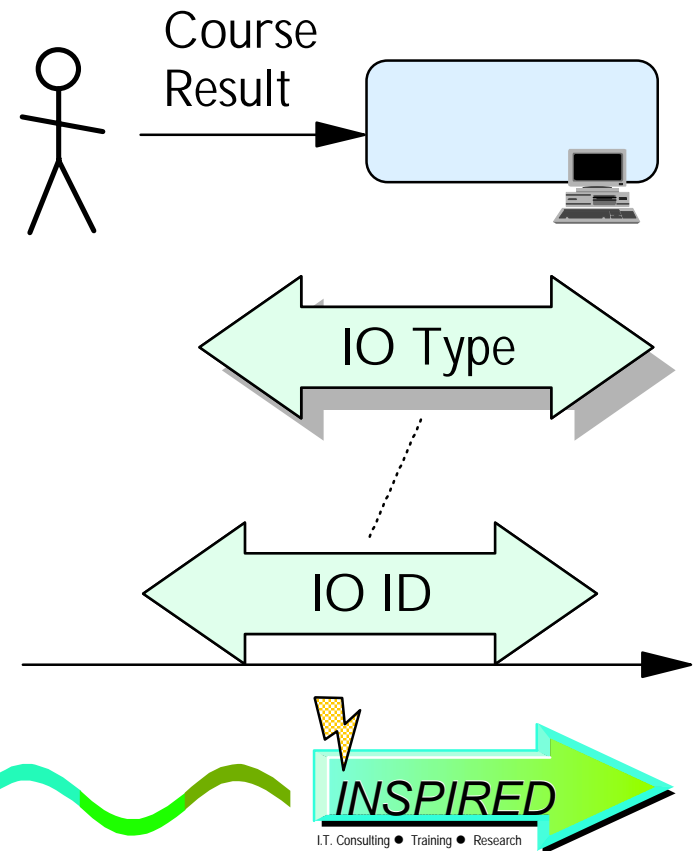
Agents (actors)

- Internal
- External

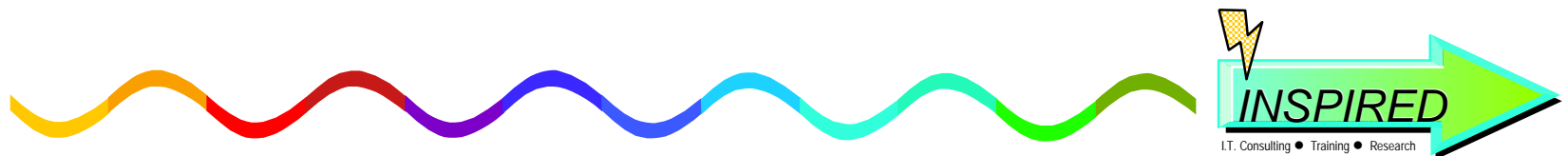
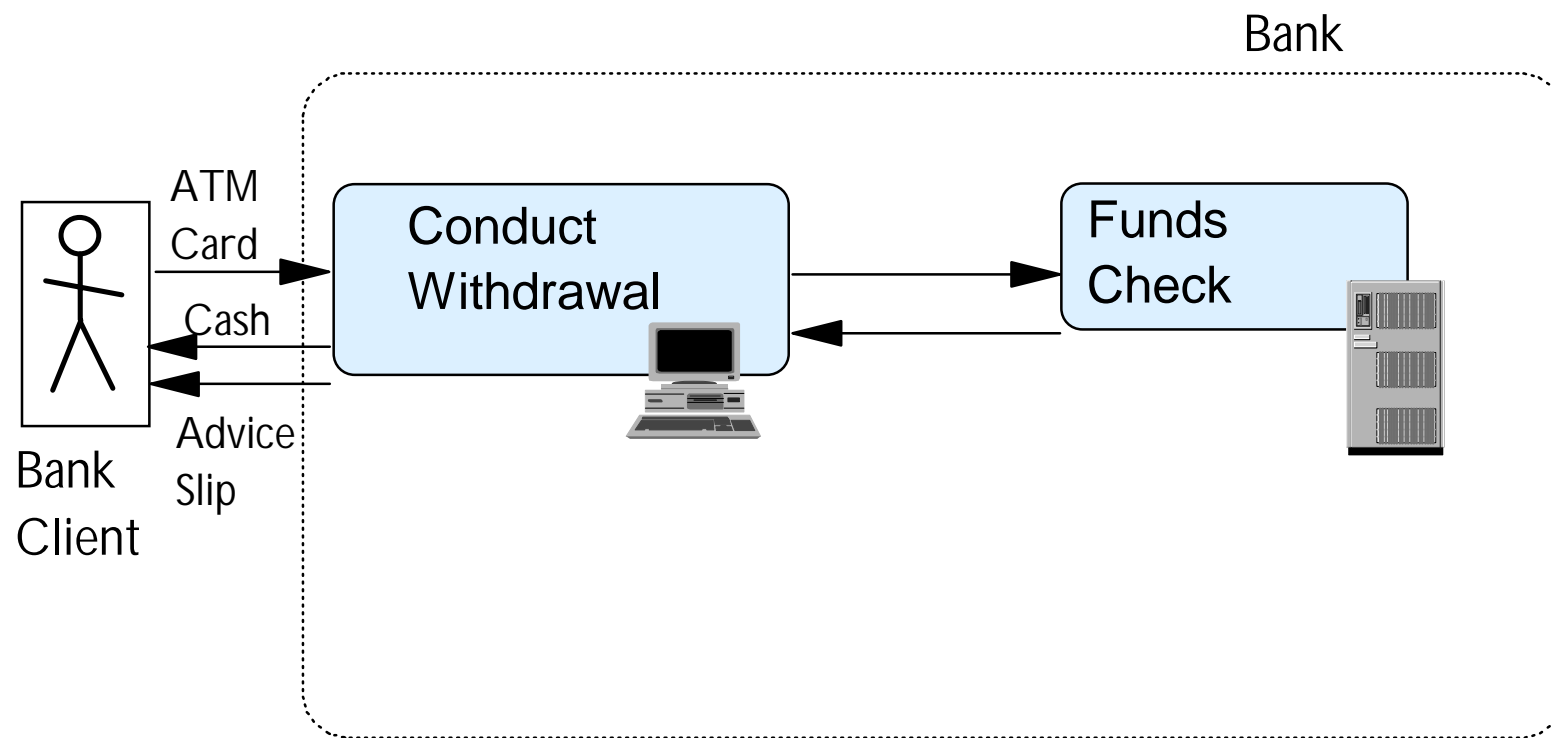


Inputs and Outputs

- Links between an Agent and an Activity normally involve input or output
- On high level models, we can just name them on the "flow"
- On more detailed models we can show them thus, and define their attributes in detail
 - Types define the medium e.g. Form, Online entry, Magnetic Card
 - IO ID references a view, which may be a simple list of attributes, or a structured prototype interface/report



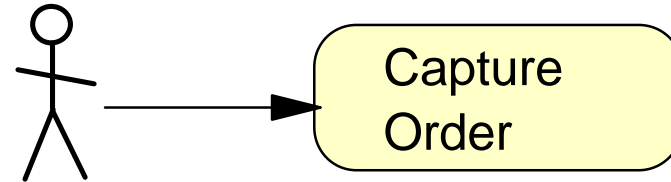
A context or use case model



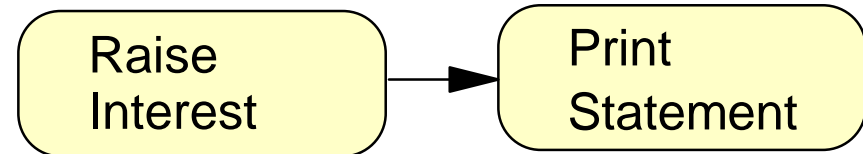
Triggers

● Activities are triggered by

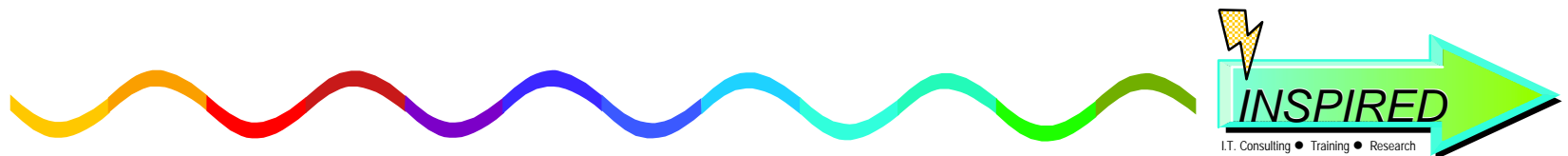
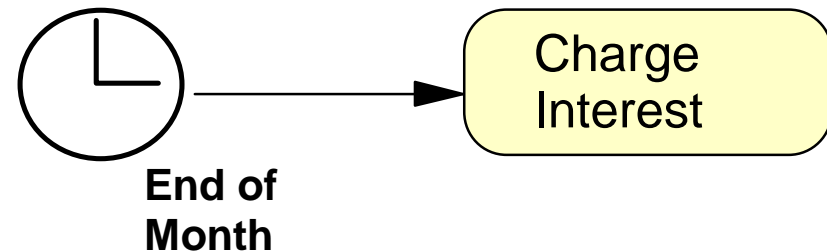
● Input from Agent



● Outcome from another activity

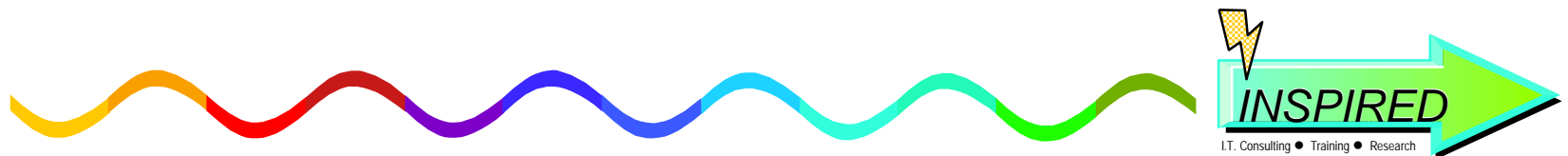
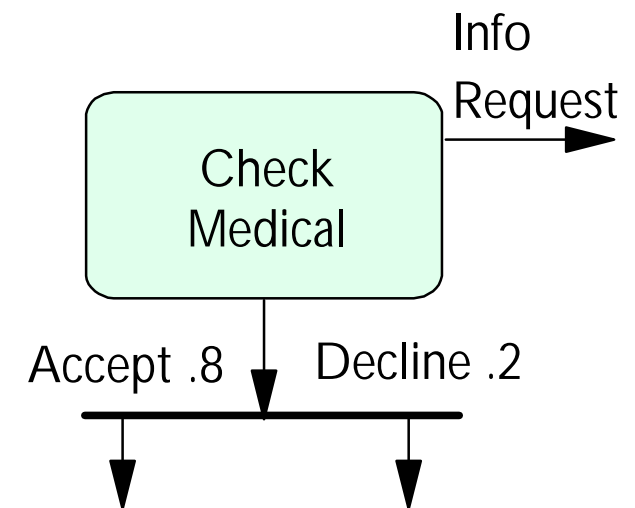
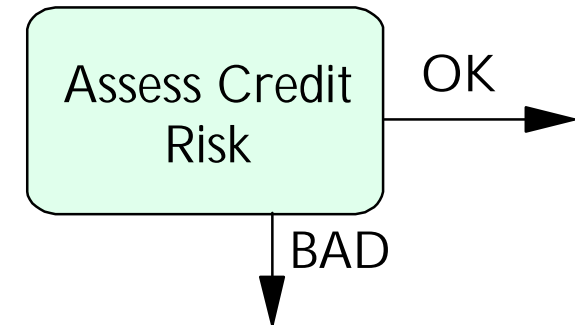


● Time (reached or elapsed)

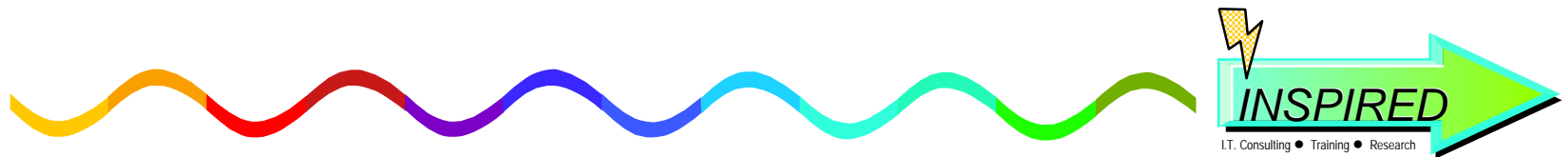
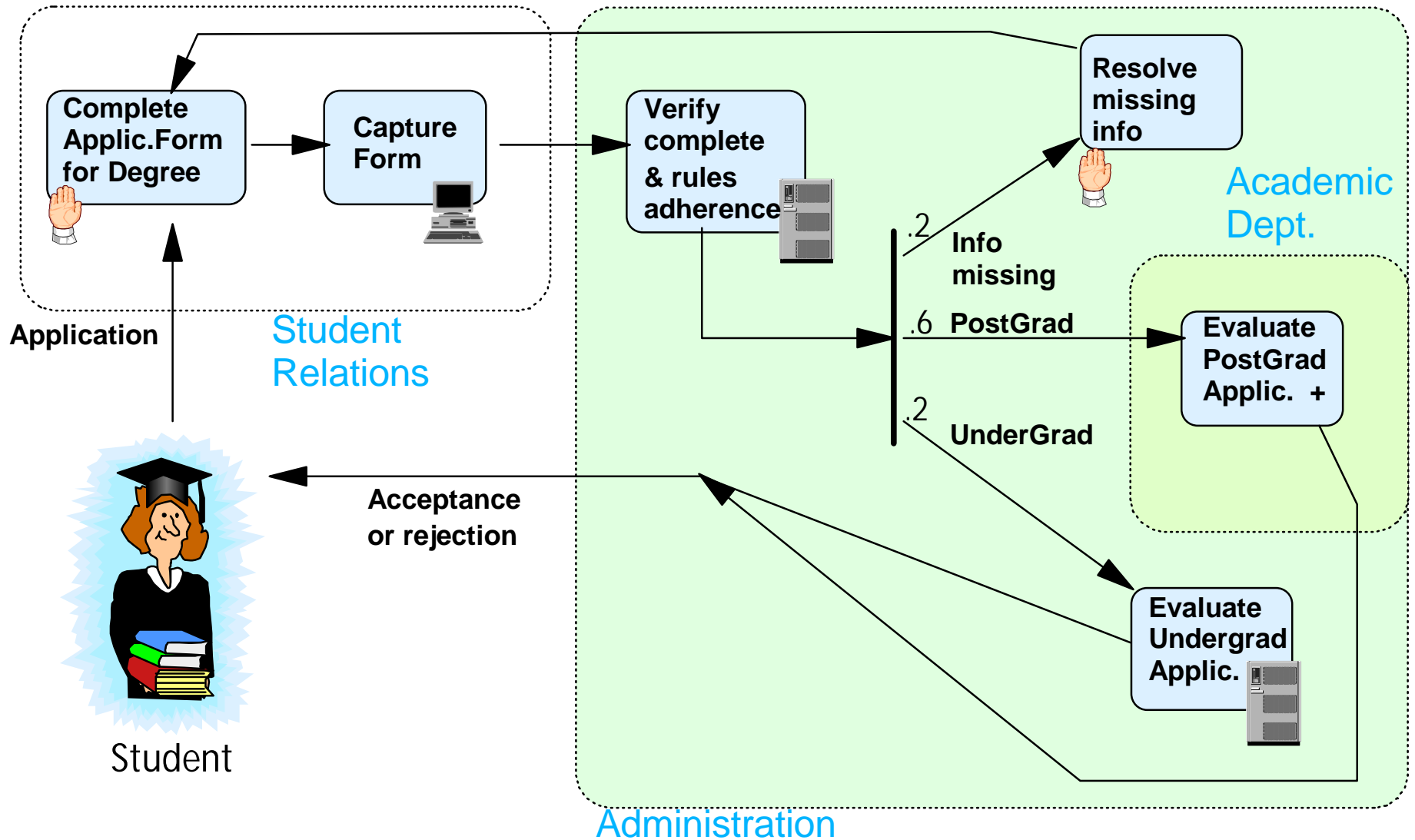


Outcomes (high level models)

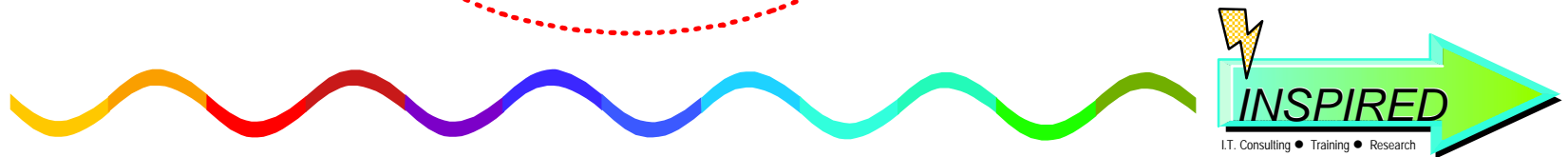
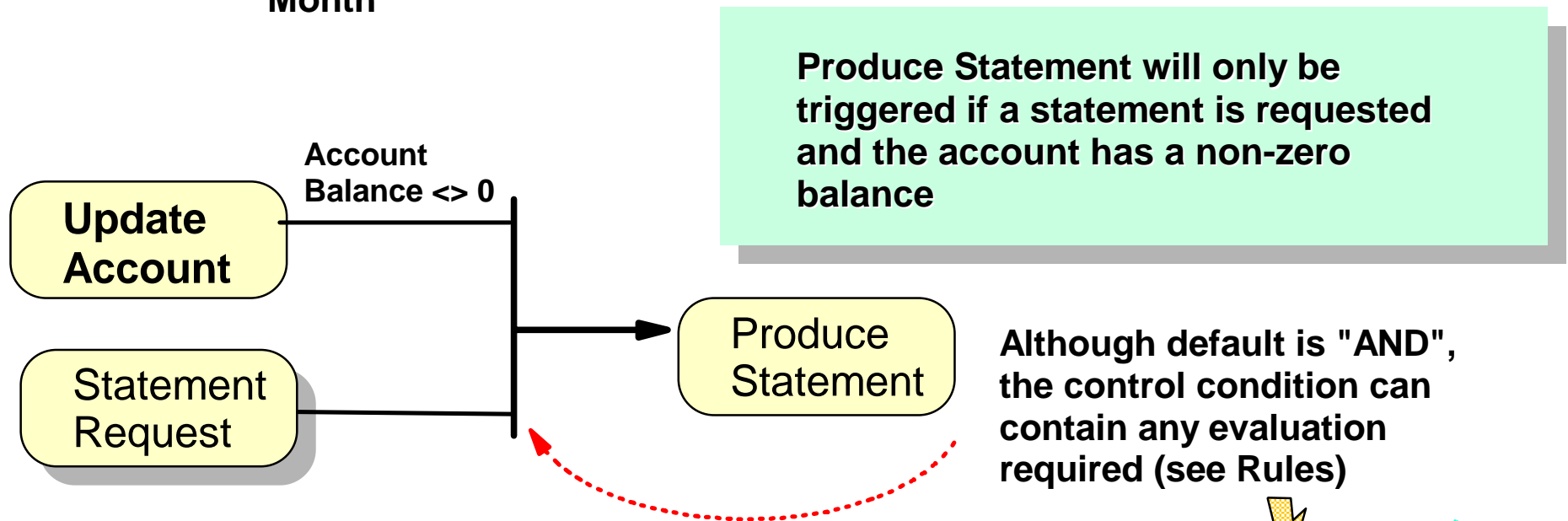
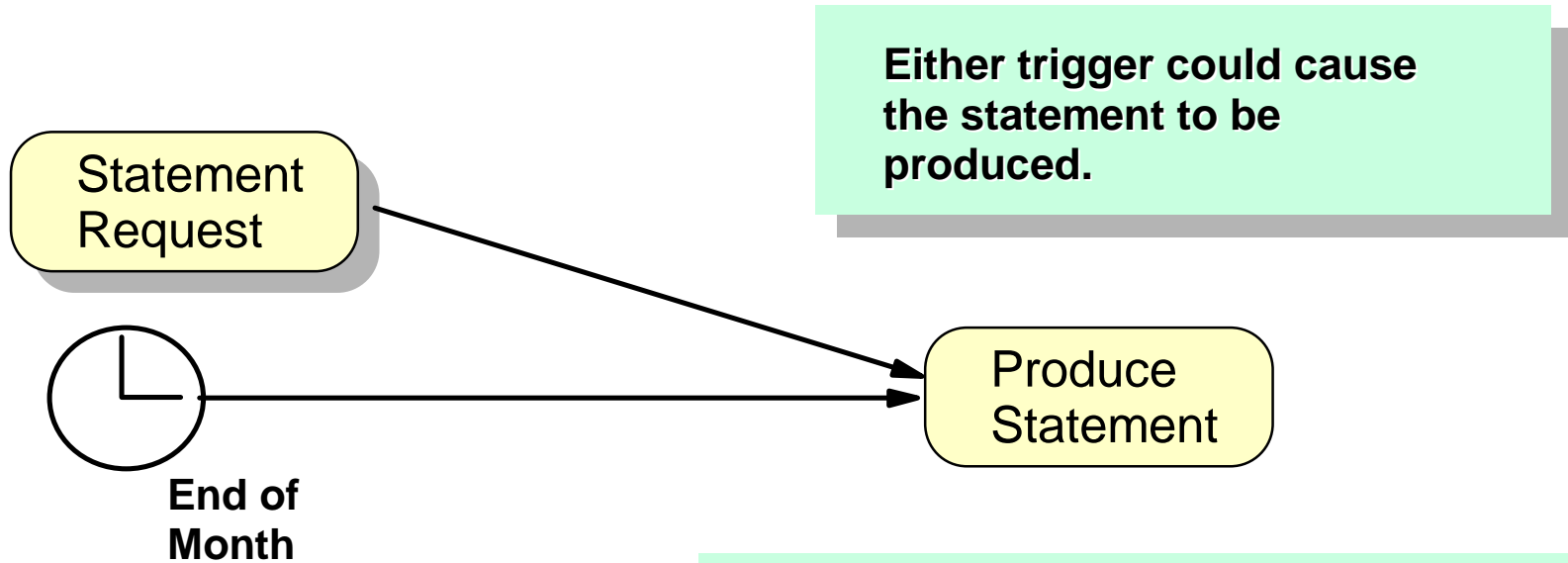
- Expressed as an annotation emerging from an Activity to distinguish flow possibilities
- Can have a probability e.g. .2 or 20%
- An activity can generate several outcomes..
- Sometimes these will be disjoint (i.e. only one of several may occur)
 - We show this with a synchronisation bar
- Outcome specification will be more rigorous later..



Business Process Model



Selective Invocation



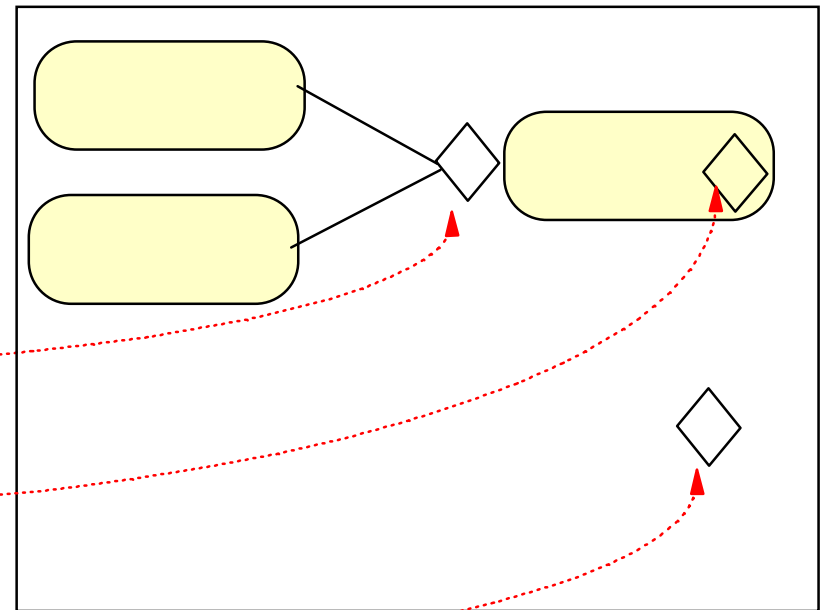
Rules

- Can be specified anywhere
- As simple text on high level models
- Or identified by diamond with reference to rule base
- Class names are highlighted
- All data items mentioned should be defined in domain model

If StockOnHand < ReorderLevel
and no PURCHASE ORDER issued
then ...

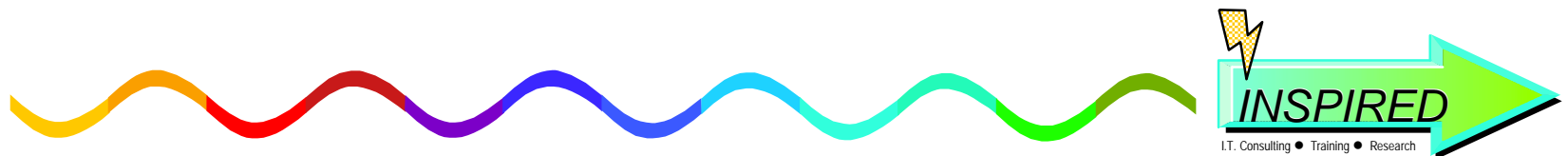
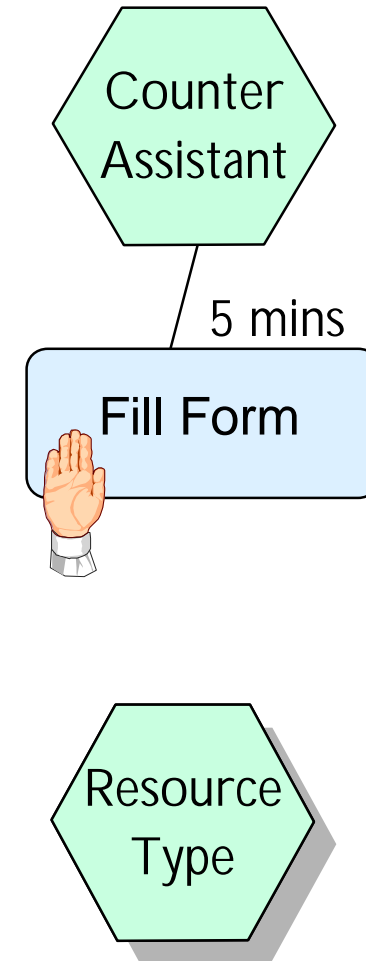
StockOnHand = PhysicalStock - CommittedStock

When QuantityOnHand < TotalDailyOrders for PRODUCT
issue to CUSTOMERS with PriorityStatus 1 first,
place BACKORDERS for CUSTOMERS not satisfied



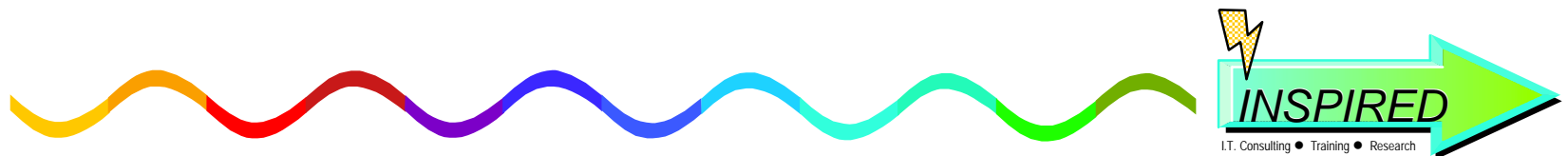
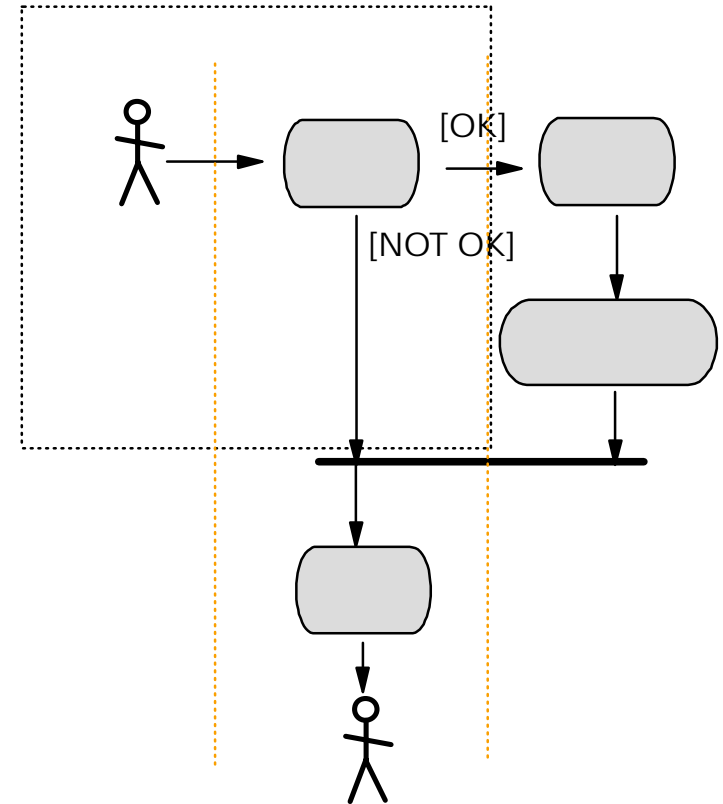
Resources

- Activities can consume resources
- Resources can be specified in activity properties
- but can also be shown on the model if desired
 - Link to activity
 - Show type if useful
 - Specify consumption per activity per pass
- Type or individual can specify number available as a constraint
 - Facilitates simulation models

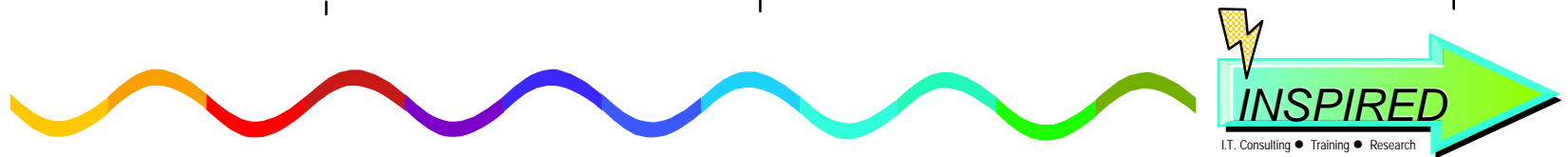
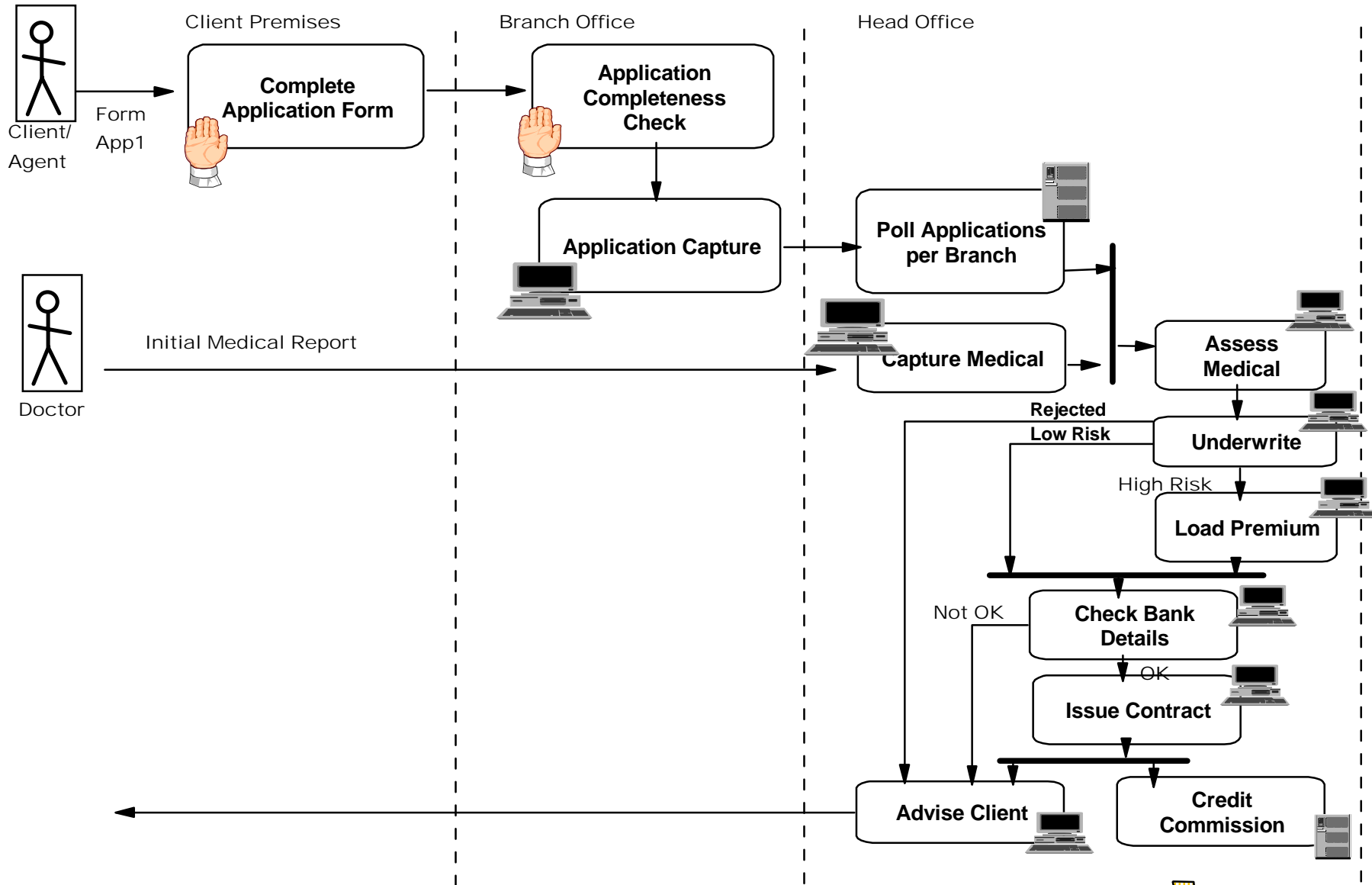


Swim lanes and bounding boxes

- Can be used to show
 - Organizational responsibility
 - Geographic location
 - Logical Transaction Start and Commit
 - Business Objects Affected (system level models)
 - Platform for deployment (design level models)
- Could have several layers or overlays per model

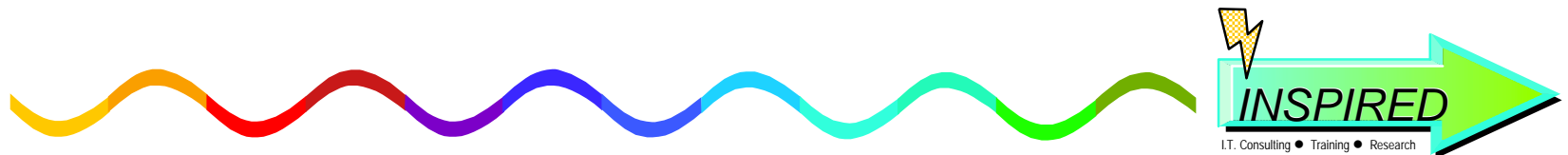


An Assurance Application

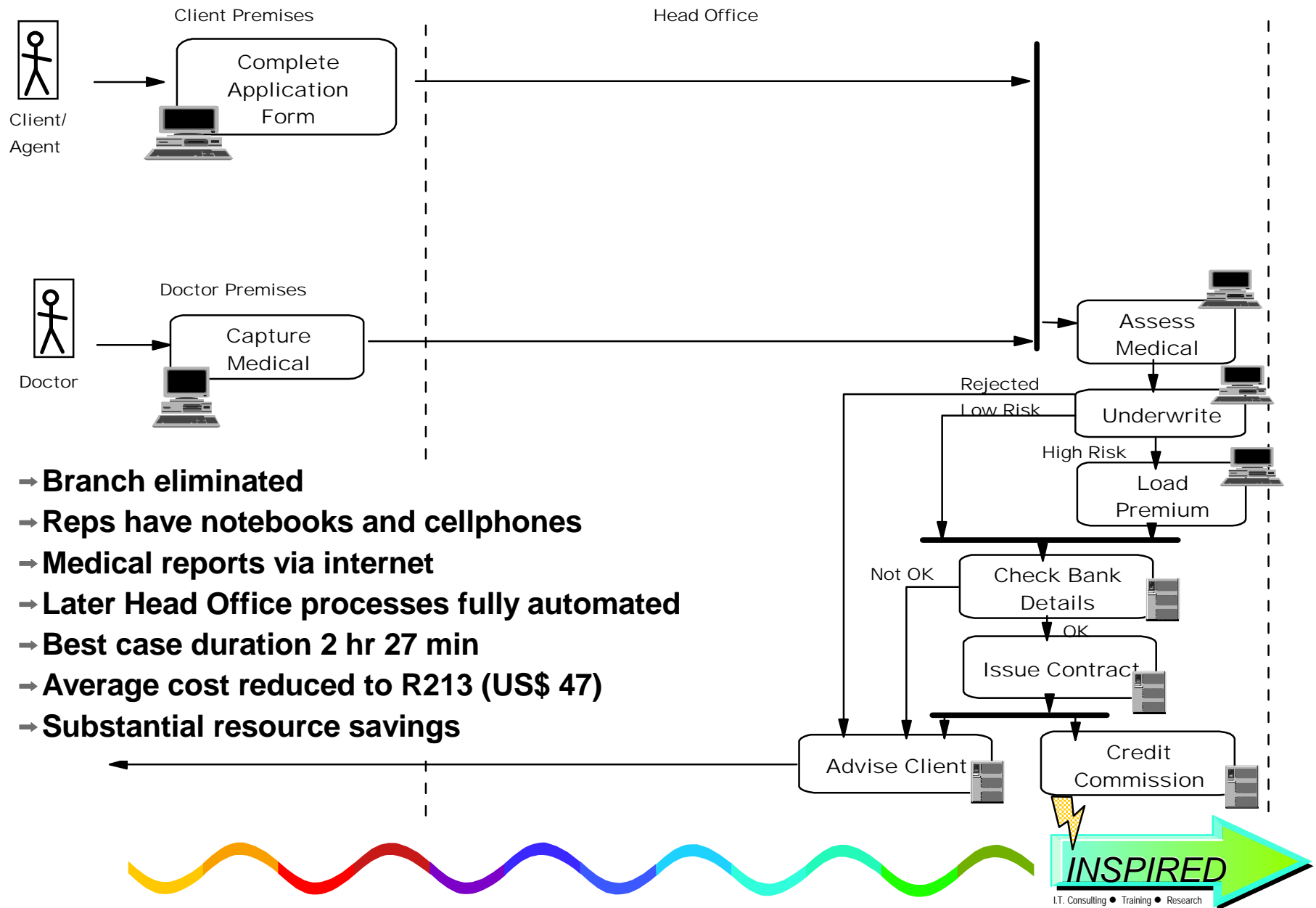


Assurance Current Process Analysis

- Best case duration 3 days 38 mins
- Average duration 3 days 1 hr 12 mins
- Average cost R326 (about US\$ 72)
- Longest delay: waiting for medical report
- No major savings by speeding up head office processes
- Fully automated functions much cheaper than computer supported ones
 - Automation will save costs
- Head office Underwriting is under staffed

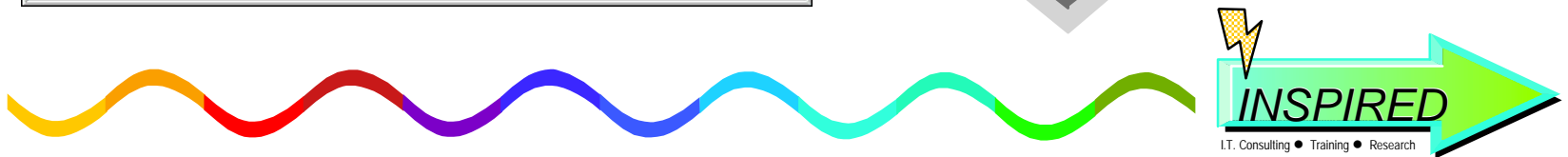
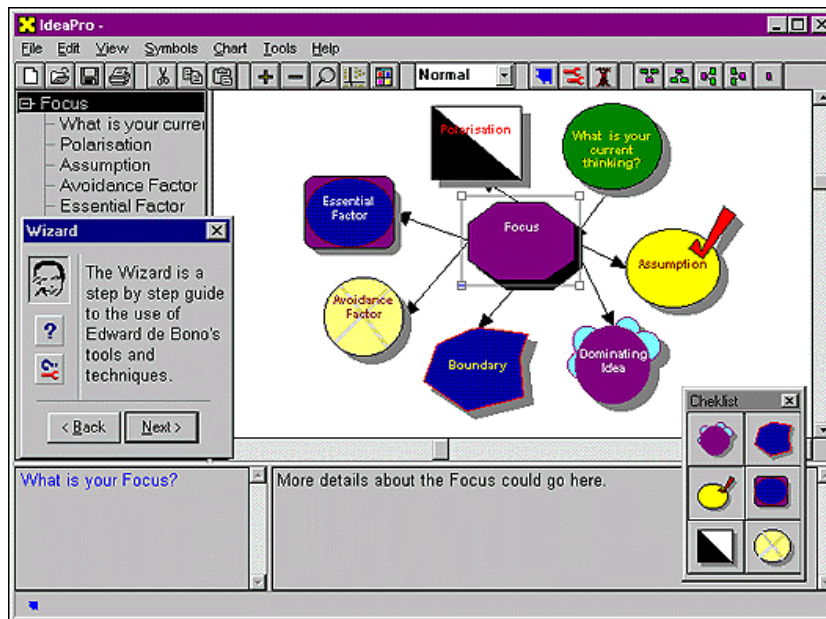


Reengineered Assurance Process



Prototype

- Refine functionality
- Define User Interface
- Refine attributes
- Clarify validation
- Verify business process flow
- Identify all exception scenarios

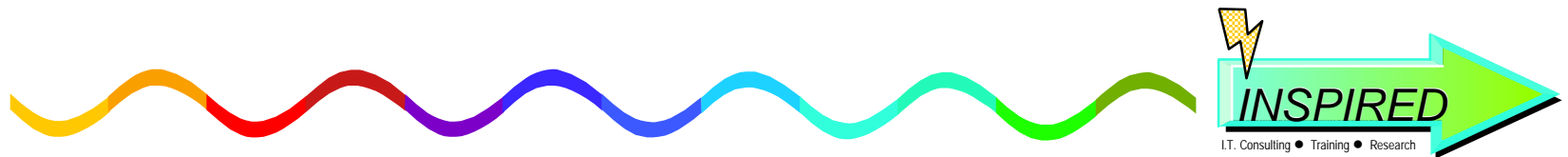
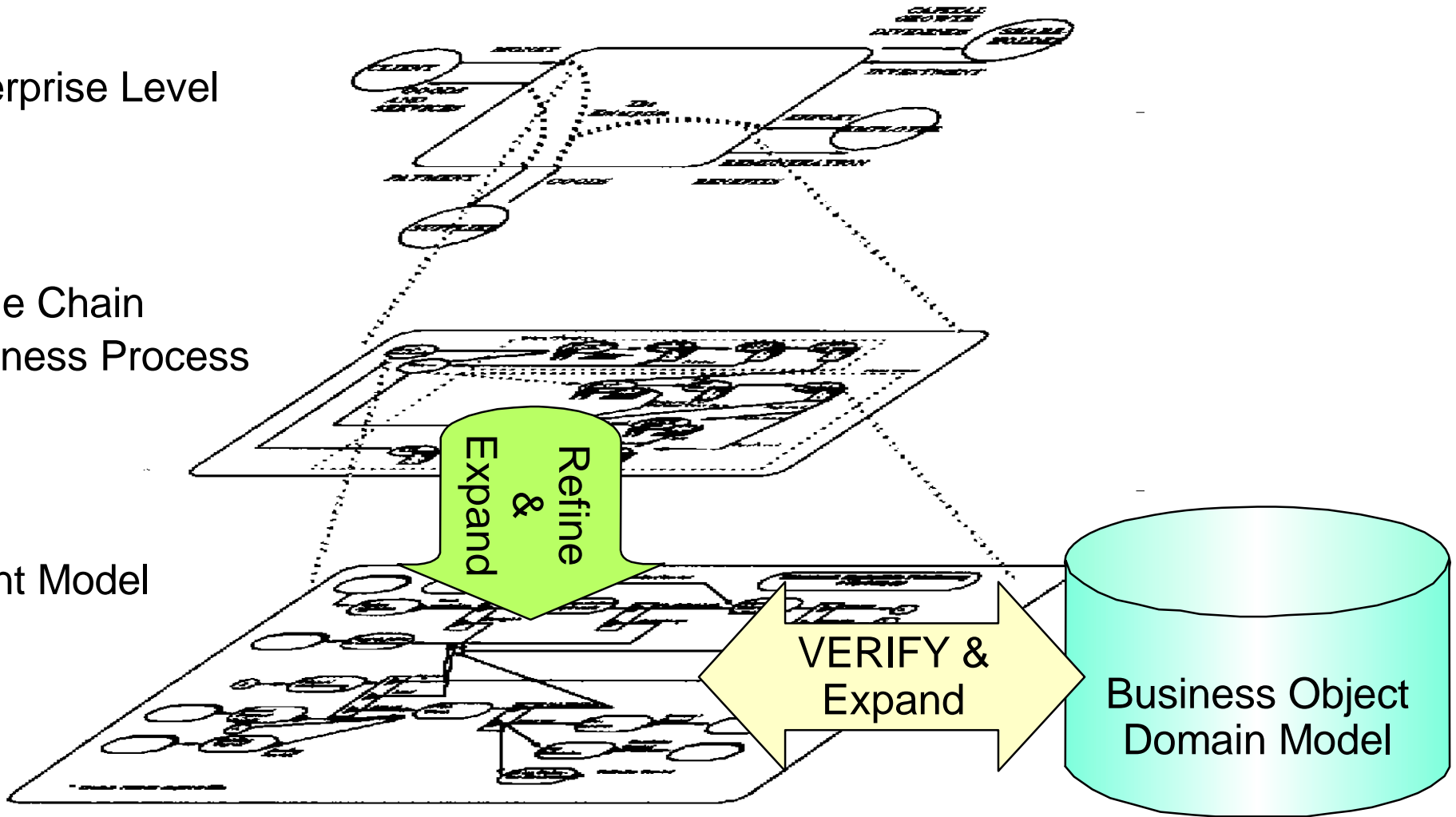


Linking Business and Event Models

Enterprise Level

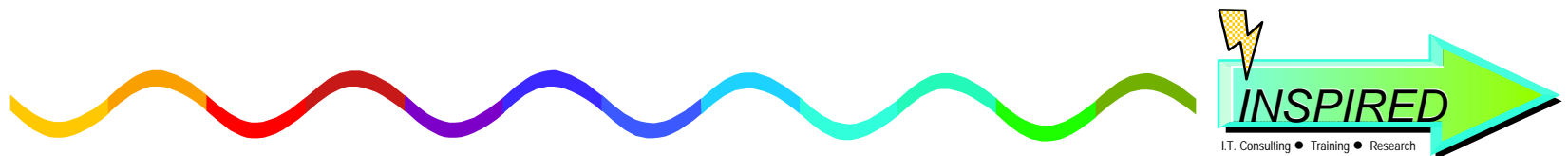
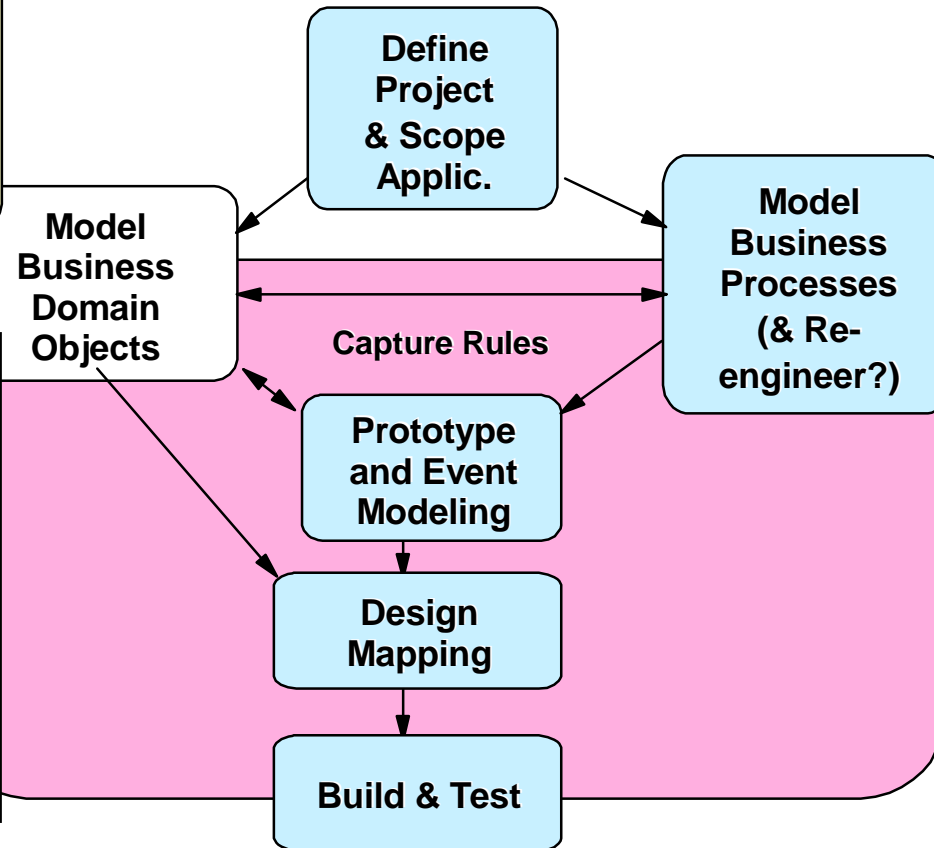
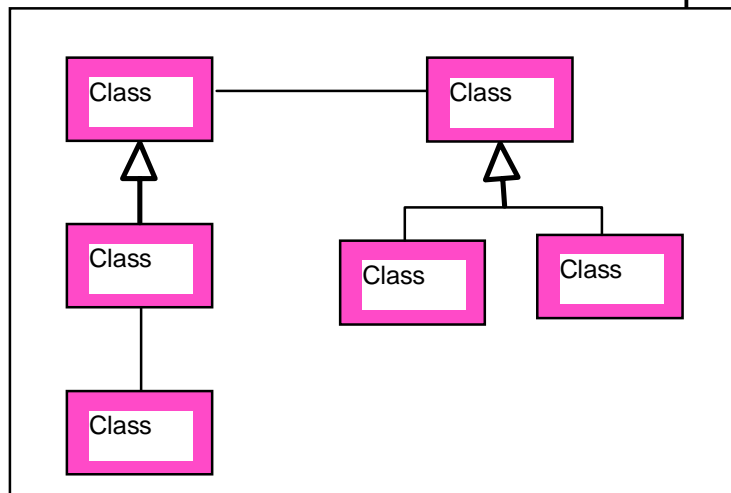
Value Chain
Business Process

Event Model

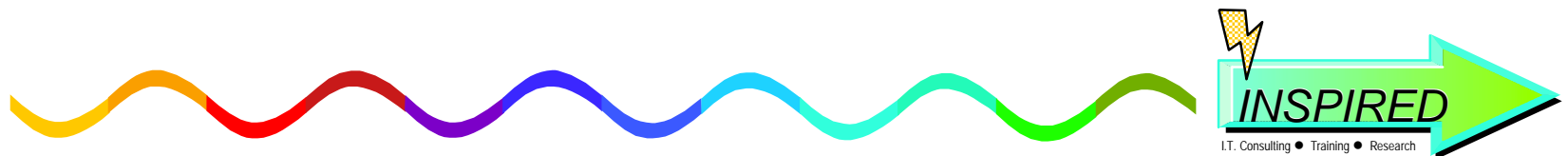
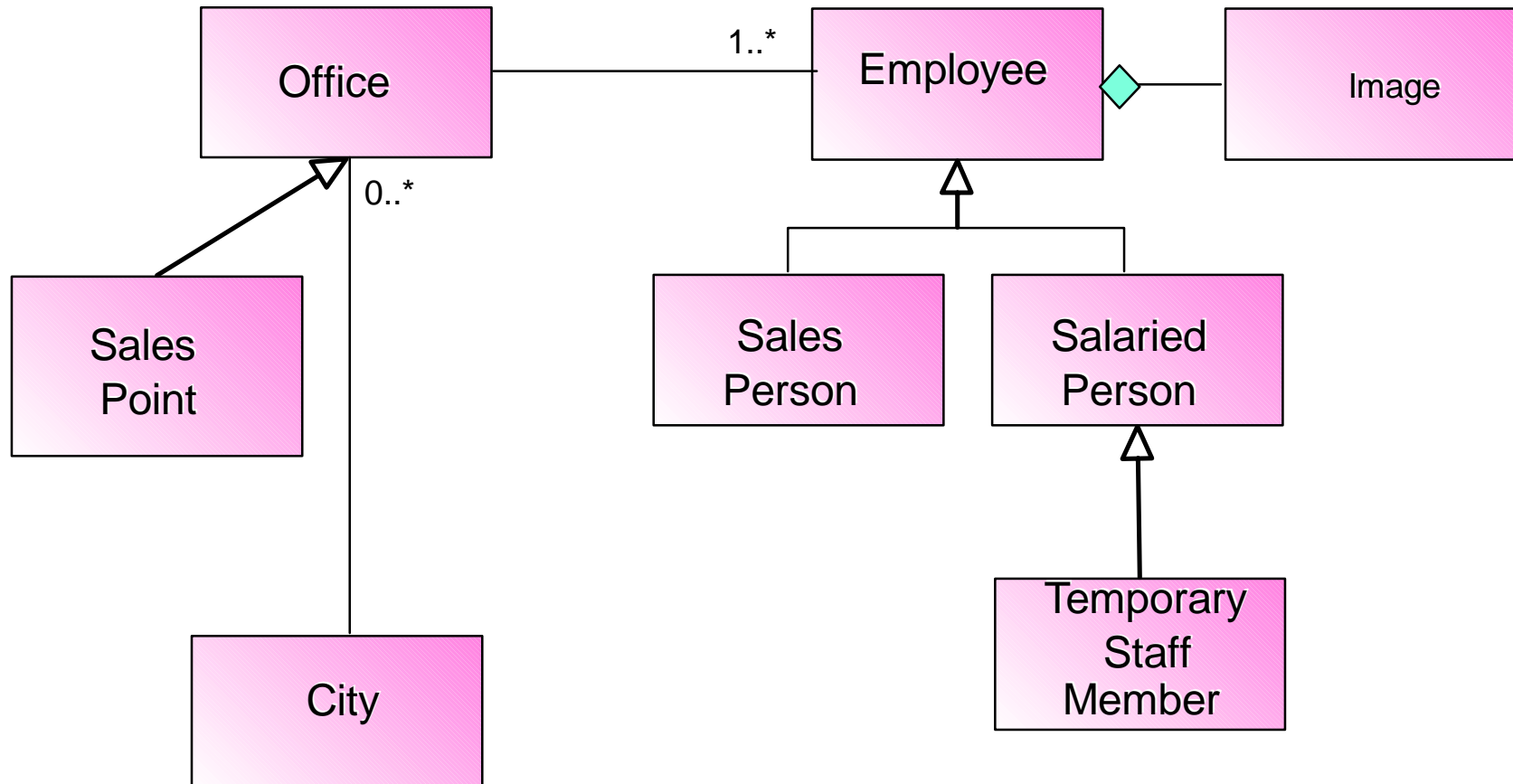


Model Business Domain Objects

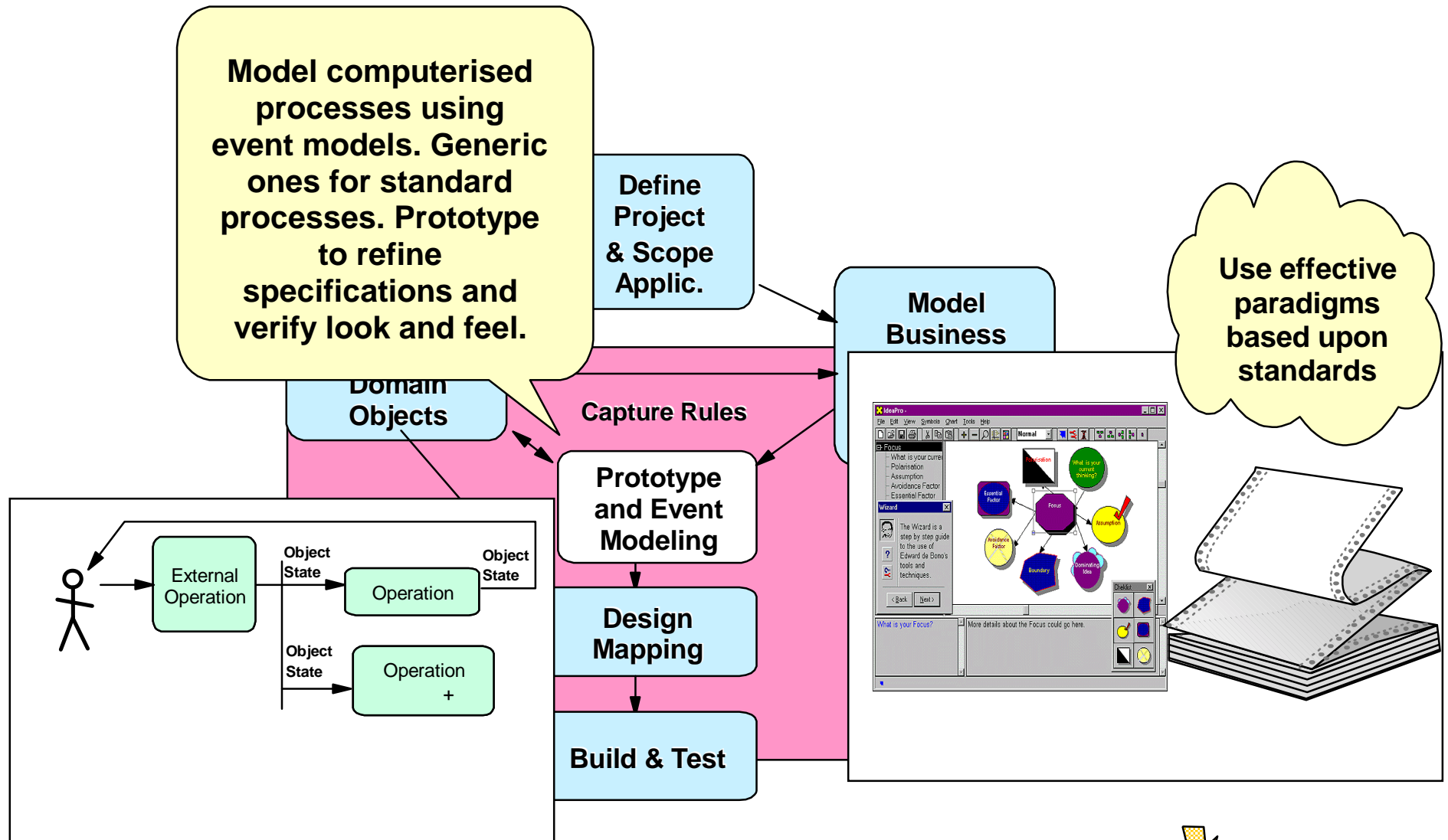
Model the entities and objects the business deals with. Record object types, relationships and attributes - no behaviours yet.
Incorporate or extend existing class definitions.



Domain Object Model (Class Diagram) Example

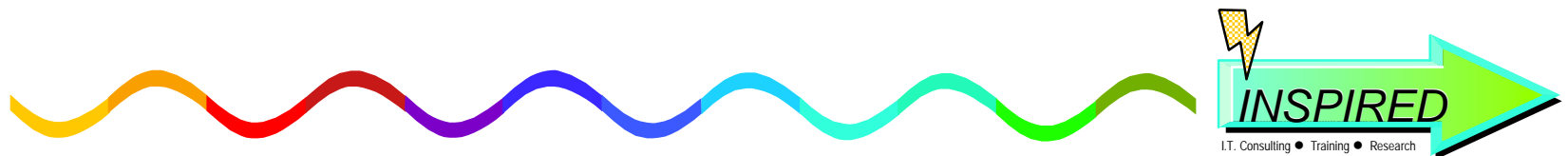
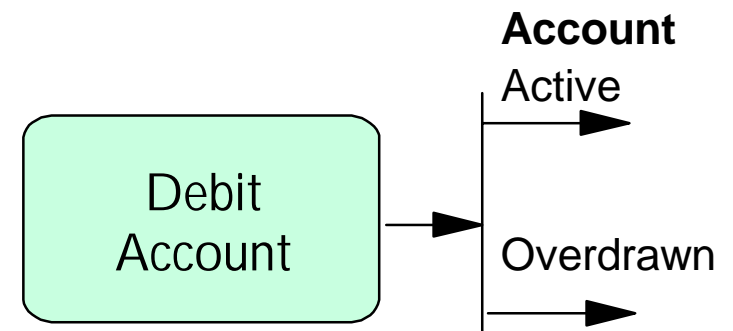
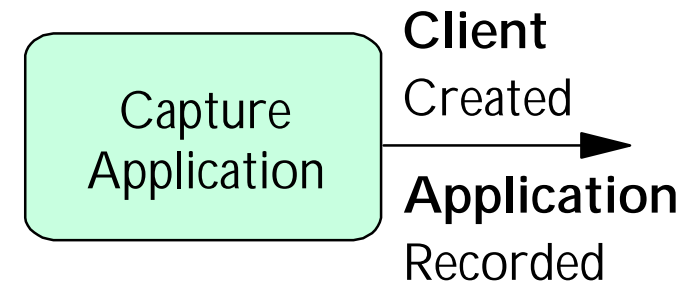


Prototyping and Event Modeling



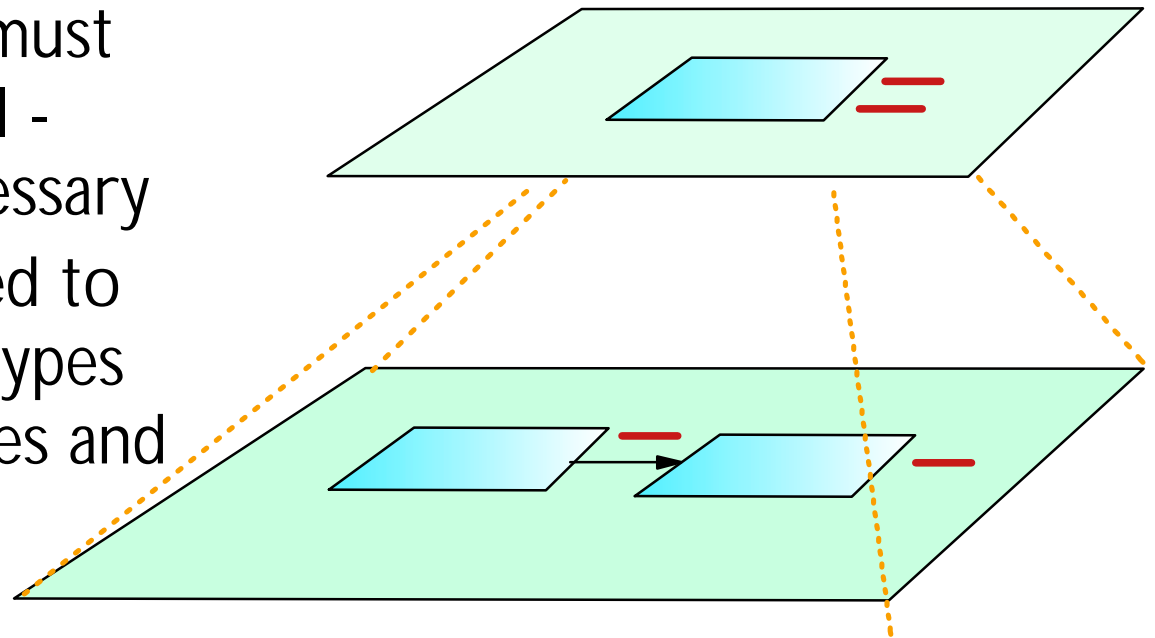
Outcomes (system level models)

- We previously recorded outcomes in an informal way
- We need to rigorously consider the effect of activities on underlying objects (domain, and at a design level, technical)
- Expressed as an annotation emerging from an Activity of the form:
[Object Type][State]
- All such effects are expressed
- Discrete states which may be reached by a given object type are shown following a synchronisation bar



Decomposition & Expansion

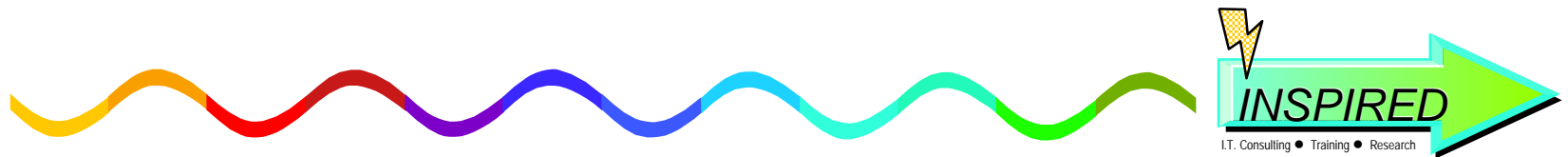
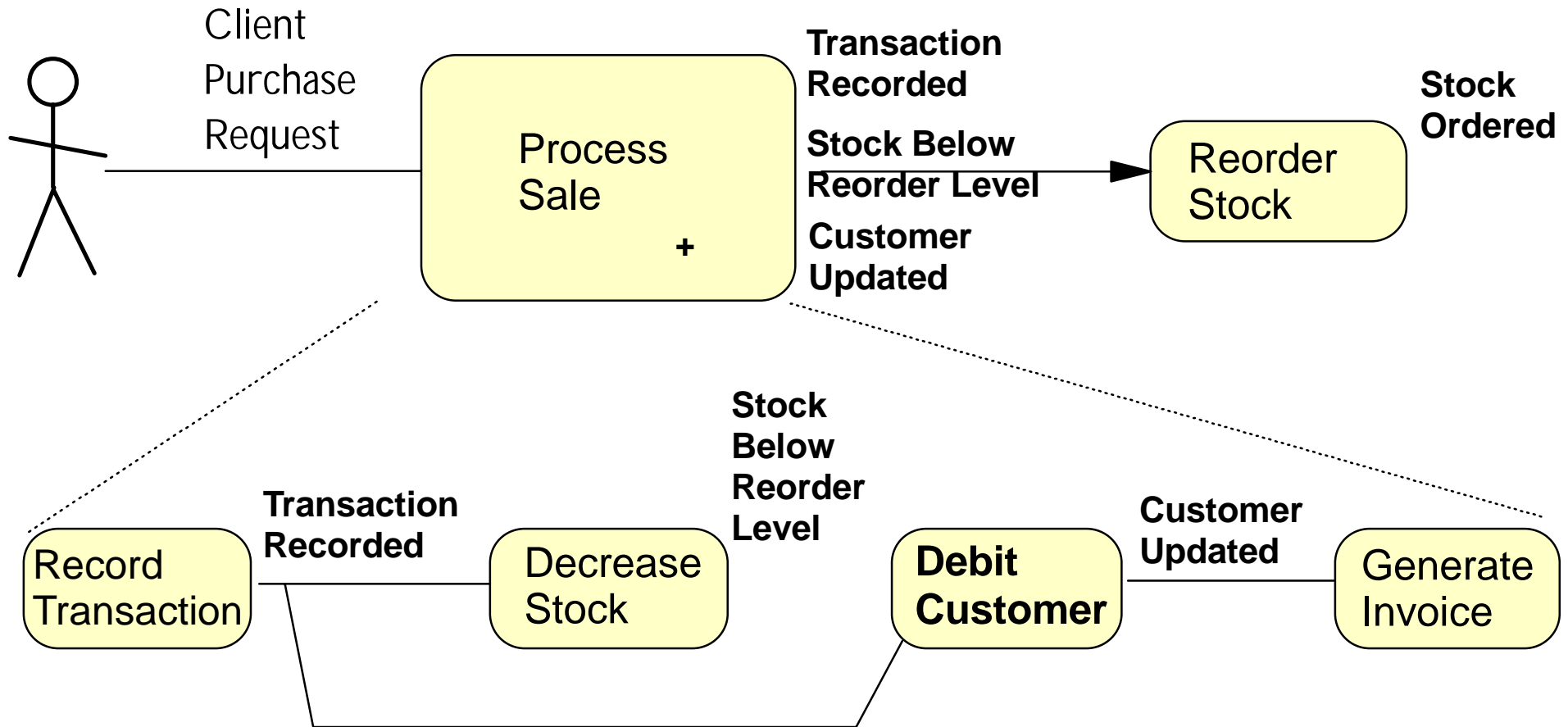
- Activities which affect the state of more than one object type are further decomposed till they do affect just one object type
- All Object Types named must exist in the domain model - update model where necessary
- Frequently, we will need to introduce new object types to manage interim states and processing



e.g. We may need a Quote before finalising an Order

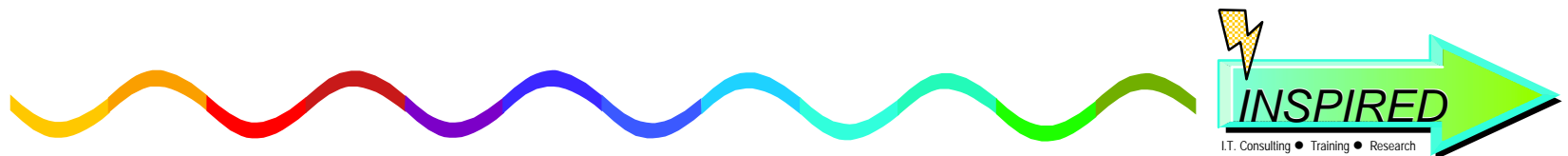
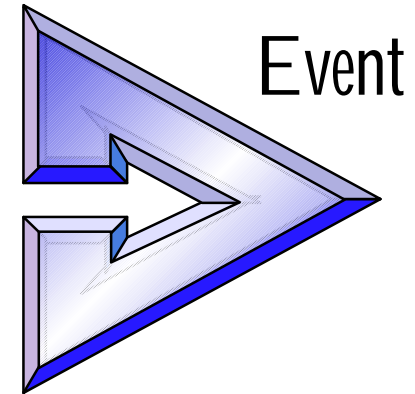


Activity decomposition example



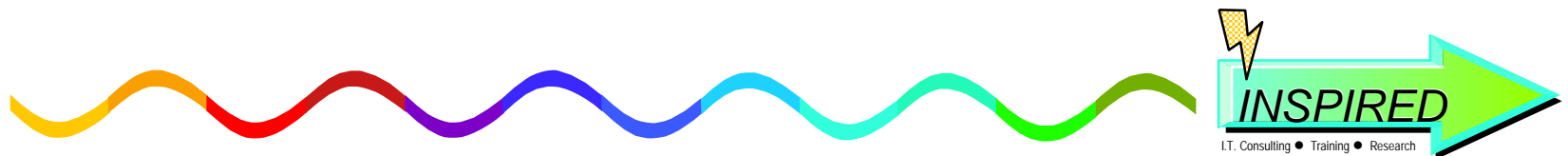
Typical Outcomes (Event Types)

- An object is *created*
- An object is *deleted*
- An object is *reclassified*
- Instances of collections are *added* or *dropped*
- The *state* of an object is changed by *updating attributes*
- An *external event* is processed or initiated
 - e.g. We get input from the screen or change the state of a device (e.g. close a communications port)

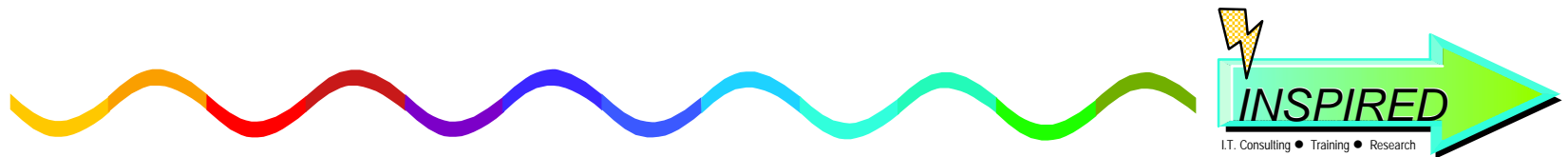
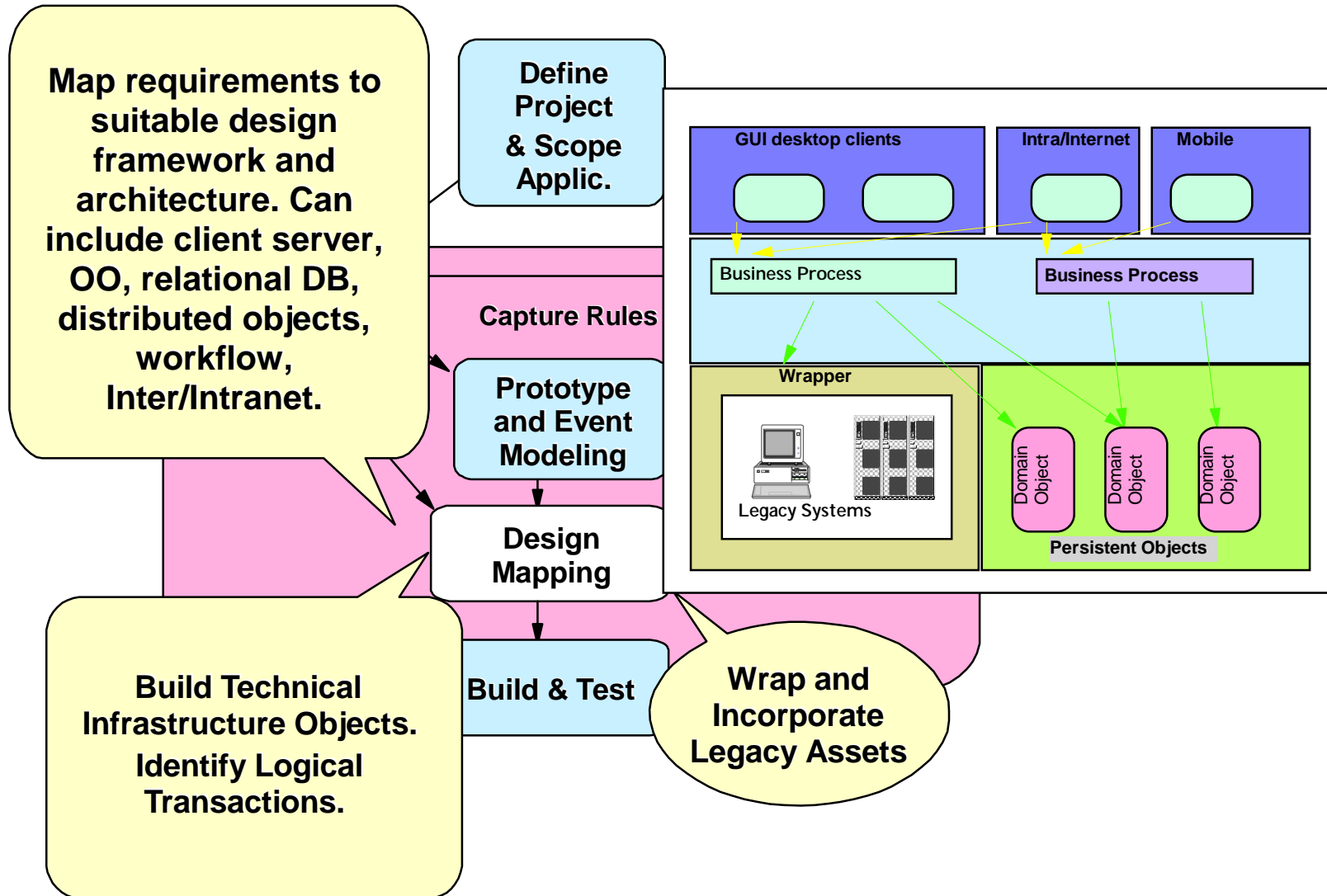


Design Level Models

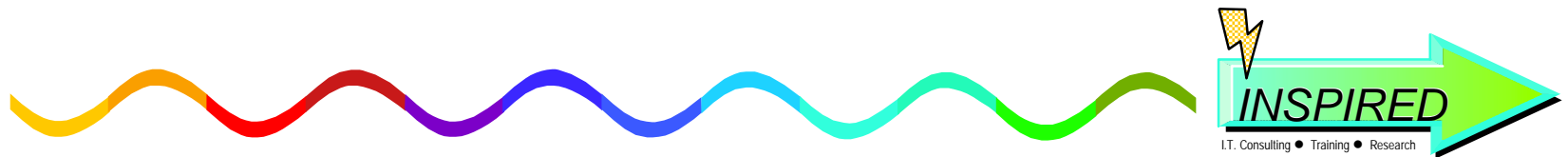
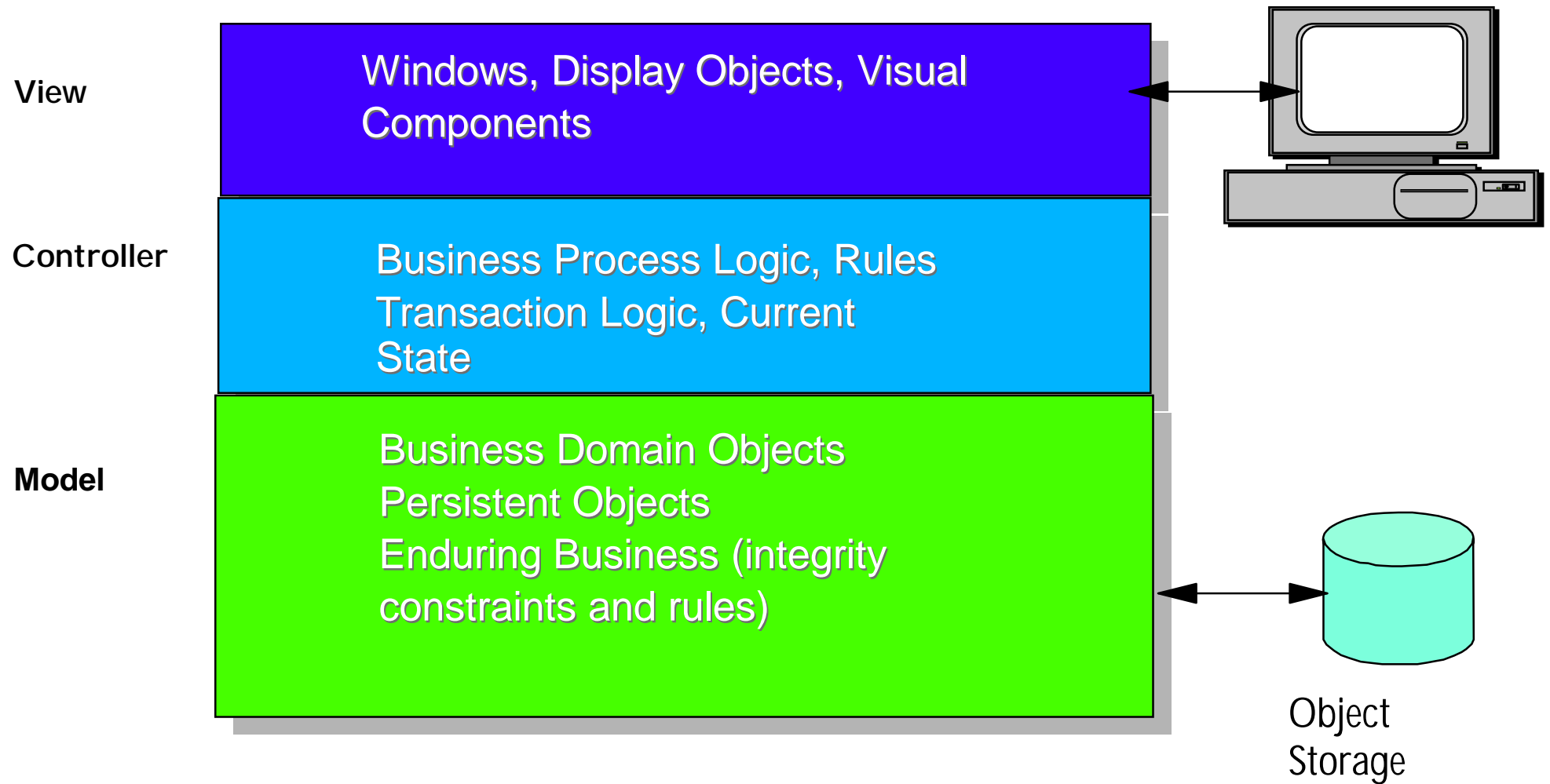
- Add technical events e.g. Logging, index maintenance...
- Show inputs and outputs to/from user interface
- Make use of frameworks and infrastructure components e.g. workflow, logging, controls
- Decompose so no activities affect more than one object type => methods
- Add logical transaction boundaries
- Can indicate platform allocation or geographic split
- With resources and volumes can be used for capacity planning, performance estimating
- Accommodate existing system and data assets



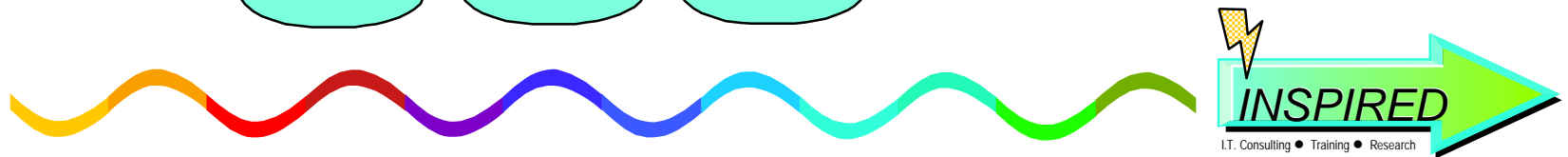
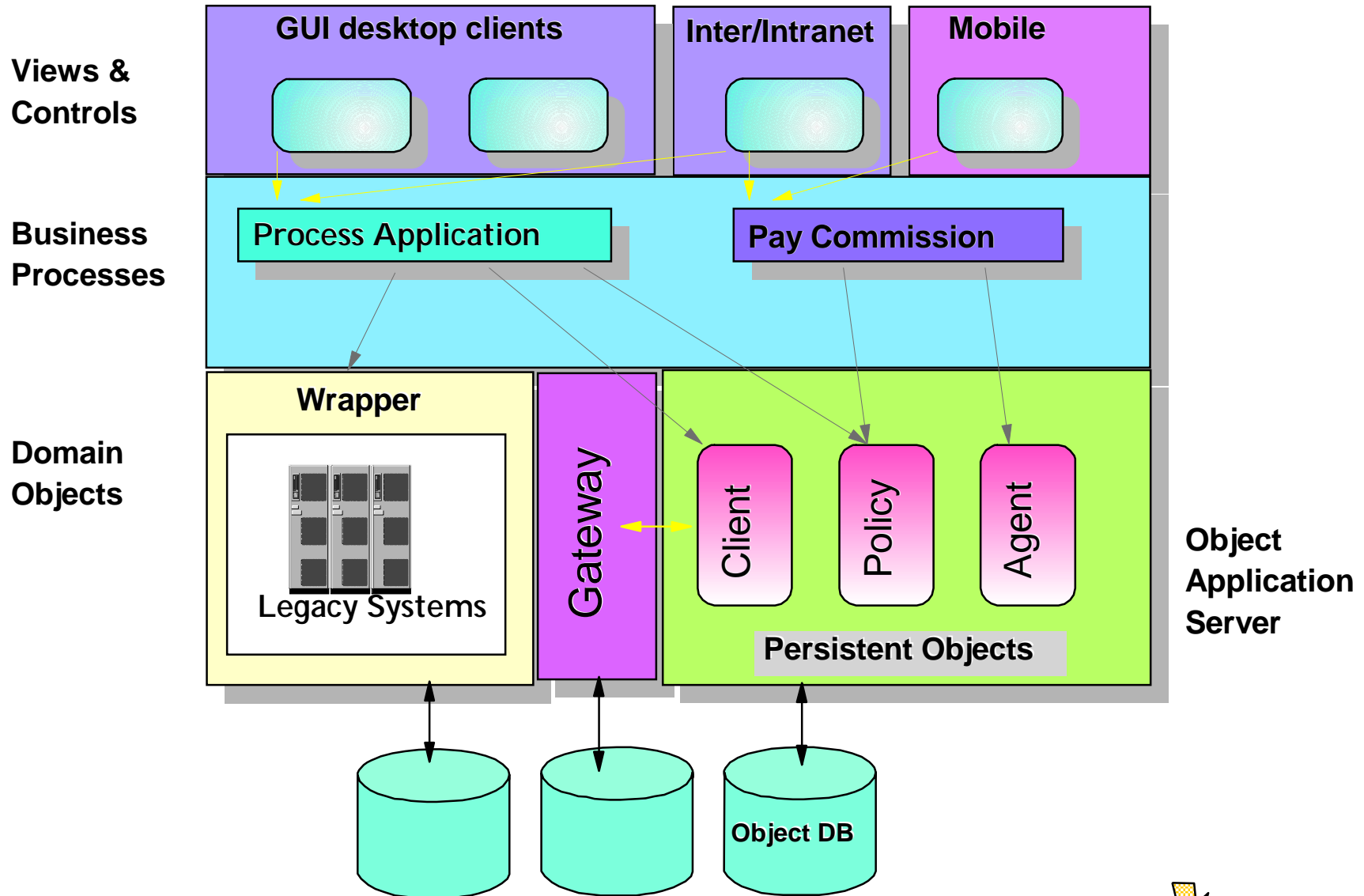
Design Mapping



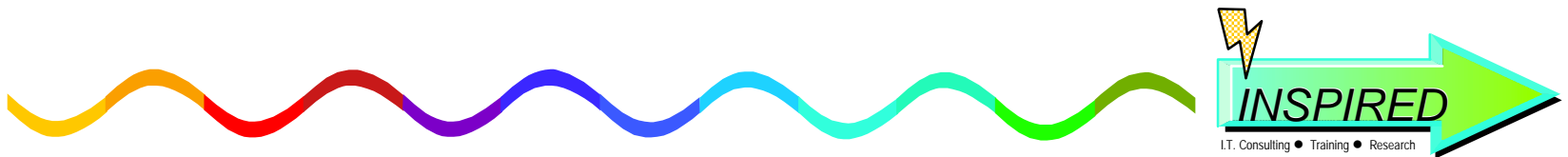
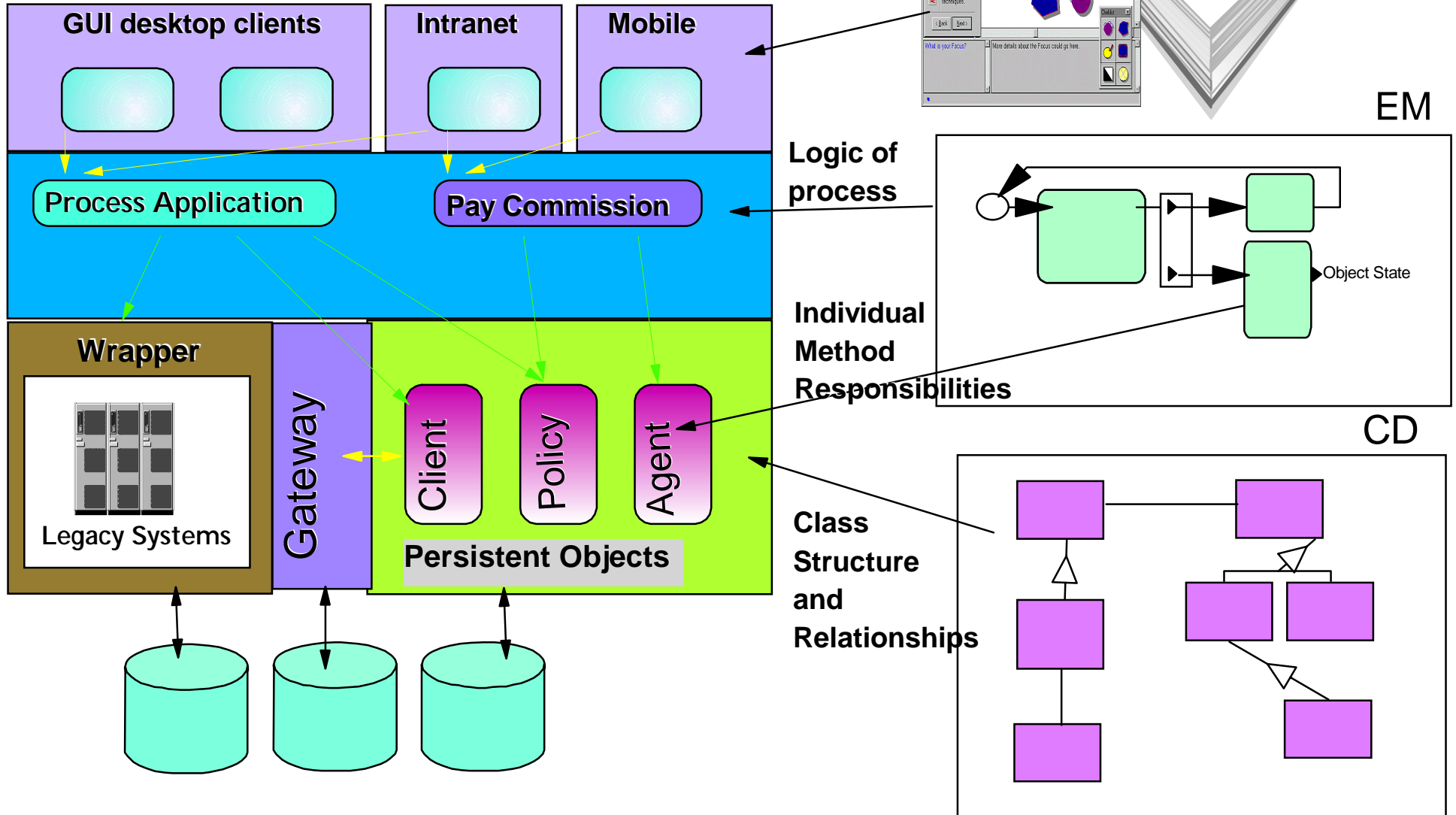
Model, View and Controller design pattern



Likely architecture of future applications

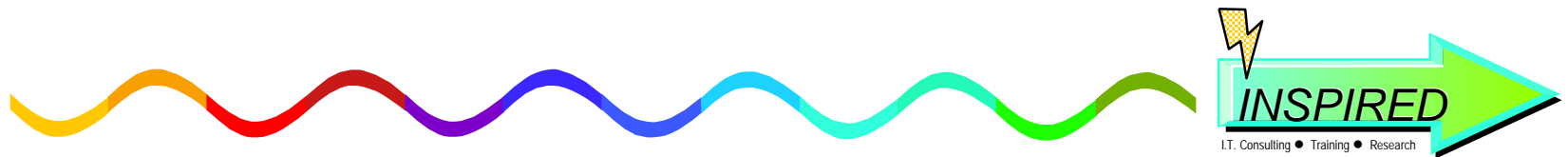
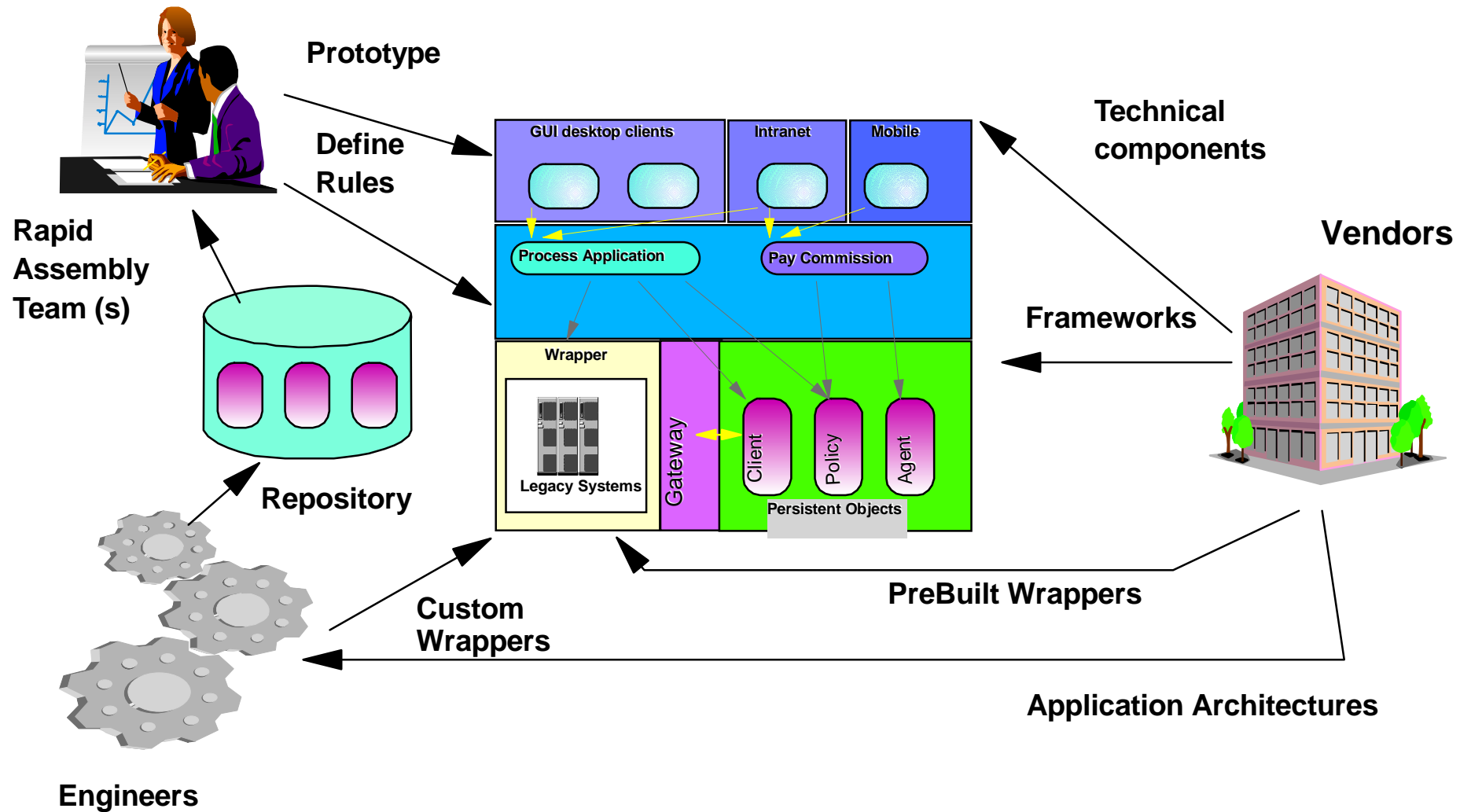


Models and the Architecture



I.T. Consulting • Training • Research

The big picture



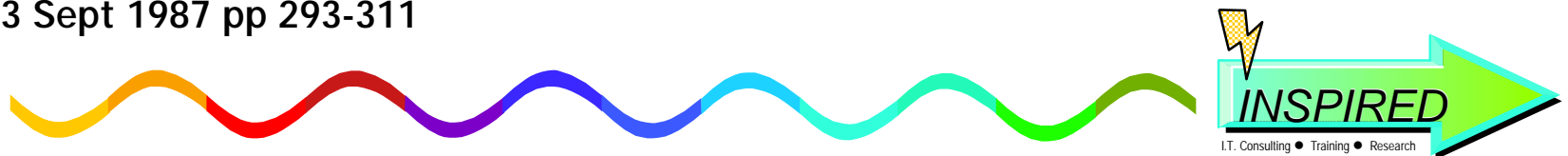
Module References

- Avison DE, Fitzgerald BA, 1995. Information Systems Development: Methodologies, Techniques and Tools. Second Edition. McGraw-Hill, 1995.
- Avison DE, Fitzgerald G, 1988. Information systems development: current themes and future directions, Information and Software Technology, October 1988.
- Avison DE, Fitzgerald G, 1986. Information Systems Development Methodologies for Microcomputer Applications, Management Forum, Sept 1986.
- Avison DE, Wood-harper AT, 1990. Multiview: An Exploration in Information Systems Development, Blackwell Scientific Publishers, Oxford 1990.
- Avison DE, Wood-harper AT, 1991. Information Systems Development Research: An Exploration of Ideas in Practice, The Computer Journal Vol 34 No 2, 1991, pp 98-112.
- Avison DE, Fitzgerald G, 1988. Information Systems Development Methodologies, Techniques and Tools, Blackwell Scientific, 1990. Booch, Grady, 1994, Object Oriented Analysis and Design with Applications (Second Edition), Benjamin Cummings
- Bourne, John, 1992, Object Oriented Engineering: Building Engineering Systems Using Smalltalk-80. Richard D Irwin, Homewood, Ill.
- Brinkkemper S, Lytinen K & Welk J (Editors), 1996, Method Engineering: Principles of method construction and tool support, Chapman & Hall, 1996.
- Butler-Cox 1988, Computer Aided Software Engineering, Butler-Cox Research Report 67, Dec 1988
- Card D N, Clark T L, Berg R A 1987, Improving software quality and productivity, Systems Vol 29 No 5 June 1987 pp 235-241
- Coad P, Yourdon E 1990, Object-Oriented Analysis, Yourdon Press 1990



Module References (cont'd)

- Colter M A 1984, A Comparative Examination of Systems Analysis Techniques, MIS Quarterly Vol 8 No 1 March 1984 pp 51-66
- Colter MA 1982, Evolution of the Structured Methodologies in Advanced Systems Development Techniques, John Wiley 1982 pp 73-96
- Comcon, 1987, The GOLD Project Management Methodology, Comcon (Pty) Ltd, 1987
- Comcon, 1987, A Project Manager's Guide to System Development Projects, Comcon (Pty) Ltd, 1987
- Downs E, Clare P & Coe I, 1988, Structured Systems Analysis and Design Method - Application and Context, Prentice Hall International, UK, 1981
- Graham, Ian, 1993, Object Oriented Methods, Addison Wesley
- Graham I, 1994. Migrating to Object Technology, Addison-Wesley, Wokingham, U.K. 1994.
- Hackathorn R D, Karimi J 1988, A Framework for Comparing Information Engineering Methods, MIS Quarterly June 1988 pp 203-220
- Humphrey, Watts S 1995, A Discipline for Software Engineering, Addison Wesley, 1995
- IEEE, 1987, Draft Standard for Software Project Management Plans - IEEE p1058, Institute of Electrical & Electronic Engineers Inc., 1987
- Infomet, 1988, The Infomet Methodology, Infomet (Pty) Ltd, 1988
- Jayaratna N, 1994. Understanding and Evaluating Methodologies. NIMSAD a Systemic Framework, McGraw-Hill, 1994.
- Jones L H, Kydd C T 1988, An Information Processing Framework for Understanding Success & Failure of MIS Methodologies, Information & Management Vol 15 1988 pp 263-271
- Mahmood M A 1987, System development methods - a comparative investigation, MIS Quarterly Vol 11 No 3 Sept 1987 pp 293-311



Module References (cont'd)

- Martin, James and Odell, James 1993, Principles of Object Oriented Analysis and Design, Prentice Hall, Englewood Cliffs, NJ
- Martin, James & Finkelstein, Clive, 1981, Information Engineering, Volume 1, Savant Research Studies, Carnforth, Lancashire, U.K.
- Martin, James, 1987, Recommended Diagramming Standards for Analysts and Programmers, Prentice Hall, Englewood Cliffs, NJ
- McLeod G, 1992. A Model for Representation, Integration and Management of Methods. South African Computer Journal.
- McLeod G, 1993. A generic method model for the Representation, Comparison, Integration and Management of Methods, Proceedings: 1st International Conference on Information Systems, Henley, U.K., 1993.
- McLeod G, 1995. MetaTool: A tool for the engineering and management of methods, Proceedings: 1st International conference on Meta-CASE. Sunderland, U.K., Jan 1995.
- Olle T W, Sol H G, Verrijn-Stuart A A, (EDS) 1982, Information Systems Design Methodologies: A Comparative Review, North Holland, Amsterdam 1982

