



Universidad de Oviedo



Software architecture and Enterprise



SOFTWARE
ARCHITECTURE

Course 2020/21

Jose E. Labra Gayo



Role of software architect

Architectural drivers

Understanding the goals
Capturing, refining and
challenging
requirements and constraints

Designing Software

Creating the technical
strategy, vision and
roadmap

Technical Risks

Identifying, mitigating and
owning the technical risks to
ensure that the architecture
"works"

Architecture evolution

Continuous technical
leadership and ownership of
the architecture throughout
The software delivery

Coding

Involvement in the
hands-on elements of
the software delivery

Quality Assurance

Introduction and adherence
to standards, guidelines,
principles, etc.

Role of software architect (review)



Expectations of an architect

Make architectural decisions

Continually analyse the architecture

Keep current with existing trends

Ensure compliance with existing decisions

Diverse exposure and experience

Have business domain knowledge

Possess interpersonal skills

Understand and navigate politics

Software architect is a role, not a rank



Understand and navigate politics

Understand the political climate of the organization
and be able to navigate the politics

Architectural decisions affect stakeholders

Product owners, project managers, business
stakeholders, developers...

Almost every decision an architect makes will be
challenged

Negotiation skills are required

Present and defend the architecture

The software architect's elevator

Communication with the different layers



Some types of companies

Product-based companies in software

- Develop some software product

- The software can itself consist of a service like Google

- The whole company is software driven

Product-based companies in other domains

- Domains like steel, textile design, logistics...

- IT department inside those companies

- Trade-off: internal IT, in-house, outsourced, offshore

Consulting or service-based companies

- Provide IT services to other companies

- From Small/local companies to International companies

Startups and entrepreneurs

- Small companies developing some product or idea

- Usually funded by angel investors or venture capitalists

- Risk in an uncertain and volatile environment



Other architects...

Enterprise architect

Support organization's business strategy with IT solutions and information

Solutions architect

Focuses on the ongoing projects and works in designing IT solutions based on requirements from the organization business

Business architect

Focuses on the organizational business needs and understands in details how the organization works

Software architect

Focuses on the ongoing project similarly to solution architects. They have a deeper knowledge in technology

Others: Data architect, application architect, technology architect,...

Enterprise architecture

Enterprise architecture

Structure and behaviors of a business

Business roles and processes

It comprises IT and organization design

2 main approaches

Zachman framework: Model driven





































TOGAF: Initiative driven



Zachman framework

Model driven approach (created by J. Zachman, 1987)

Classification scheme for enterprise descriptions

	WHAT	HOW	WHERE	WHO	WHEN	WHY	
SCOPE CONTEXTS	<div>Inventory Identification</div> <div></div> <div>Inventory Types</div>	<div>Process Identification</div> <div></div> <div>Process Types</div>	<div>Network Identification</div> <div></div> <div>Network Types</div>	<div>Organization Identification</div> <div></div> <div>Organization Types</div>	<div>Timing Identification</div> <div></div> <div>Timing Types</div>	<div>Motivation Identification</div> <div></div> <div>Motivation Types</div>	STRATEGISTS AS THEORISTS
BUSINESS CONCEPTS	<div>Inventory Definition</div> <div></div> <div>Business Entity Business Relationship</div>	<div>Process Definition</div> <div></div> <div>Business Transform Business Input</div>	<div>Network Definition</div> <div></div> <div>Business Location Business Connection</div>	<div>Organization Definition</div> <div></div> <div>Business Role Business Work</div>	<div>Timing Definition</div> <div></div> <div>Business Cycle Business Moment</div>	<div>Motivation Definition</div> <div></div> <div>Business End Business Means</div>	EXECUTIVE LEADERS AS OWNERS
SYSTEM LOGIC	<div>Inventory Representation</div> <div></div> <div>System Entity System Relationship</div>	<div>Process Representation</div> <div></div> <div>System Transform System Input</div>	<div>Network Representation</div> <div></div> <div>System Location System Connection</div>	<div>Organization Representation</div> <div></div> <div>System Role System Work</div>	<div>Timing Representation</div> <div></div> <div>System Cycle System Moment</div>	<div>Motivation Representation</div> <div></div> <div>System End System Means</div>	ARCHITECTS AS DESIGNERS
TECHNOLOGY PHYSICS	<div>Inventory Specification</div> <div></div> <div>Technology Entity Technology Relationship</div>	<div>Process Specification</div> <div></div> <div>Technology Transform Technology Input</div>	<div>Network Specification</div> <div></div> <div>Technology Location Technology Connection</div>	<div>Organization Specification</div> <div></div> <div>Technology Role Technology Work</div>	<div>Timing Specification</div> <div></div> <div>Technology Cycle Technology Moment</div>	<div>Motivation Specification</div> <div></div> <div>Technology End Technology Means</div>	ENGINEERS AS BUILDERS
COMPONENT ASSEMBLIES	<div>Inventory Configuration</div> <div></div> <div>Component Entity Component Relationship</div>	<div>Process Configuration</div> <div></div> <div>Component Transform Component Input</div>	<div>Network Configuration</div> <div></div> <div>Component Location Component Connection</div>	<div>Organization Configuration</div> <div></div> <div>Component Role Component Work</div>	<div>Timing Configuration</div> <div></div> <div>Component Cycle Component Moment</div>	<div>Motivation Configuration</div> <div></div> <div>Component End Component Means</div>	TECHNICIANS AS IMPLEMENTERS
OPERATIONS CLASSES	<div>Inventory Instantiation</div> <div></div> <div>Operations Entity Operations Relationship</div>	<div>Process Instantiation</div> <div></div> <div>Operations Transform Operations Input</div>	<div>Network Instantiation</div> <div></div> <div>Operations Location Operations Connection</div>	<div>Organization Instantiation</div> <div></div> <div>Operations Role Operations Work</div>	<div>Timing Instantiation</div> <div></div> <div>Operations Cycle Operations Moment</div>	<div>Motivation Instantiation</div> <div></div> <div>Operations End Operations Means</div>	WORKERS AS PARTICIPANTS
	INVENTORY SETS	PROCESS TRANSFORMATIONS	NETWORK NODES	ORGANIZATION GROUPS	TIMING PERIODS	MOTIVATION REASONS	

TOGAF (The Open Group Architecture Framework)

Initiative driven approach

Framework and methodology

First published in 1995, Dpt. Defense USA

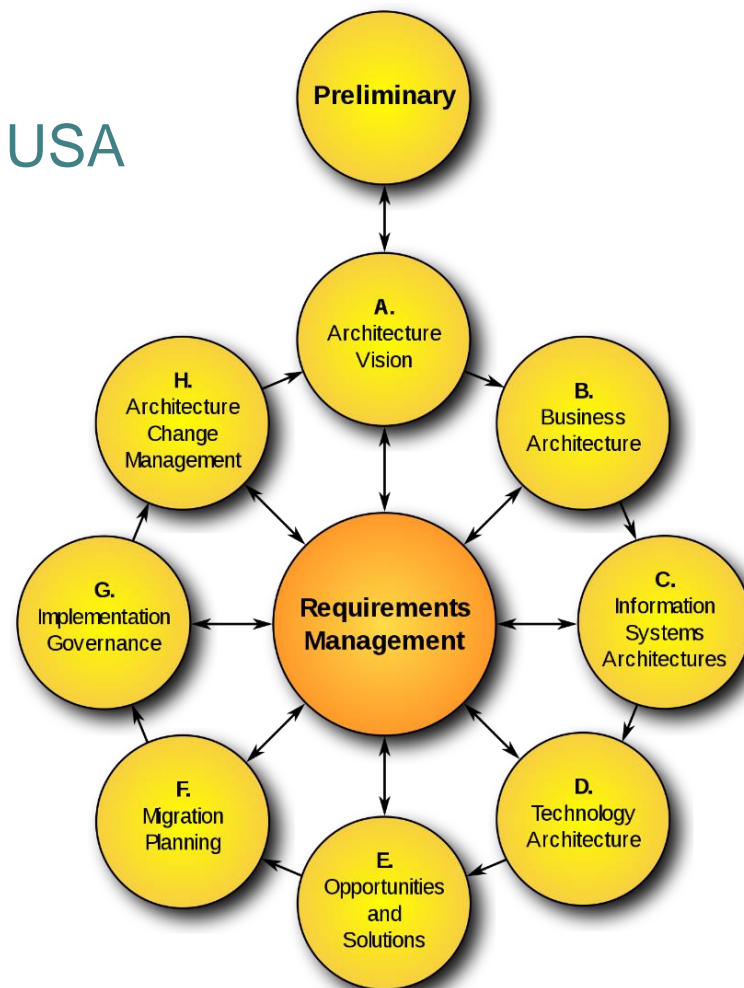
4 architecture domains

Business architecture

Data architecture

Applications architecture

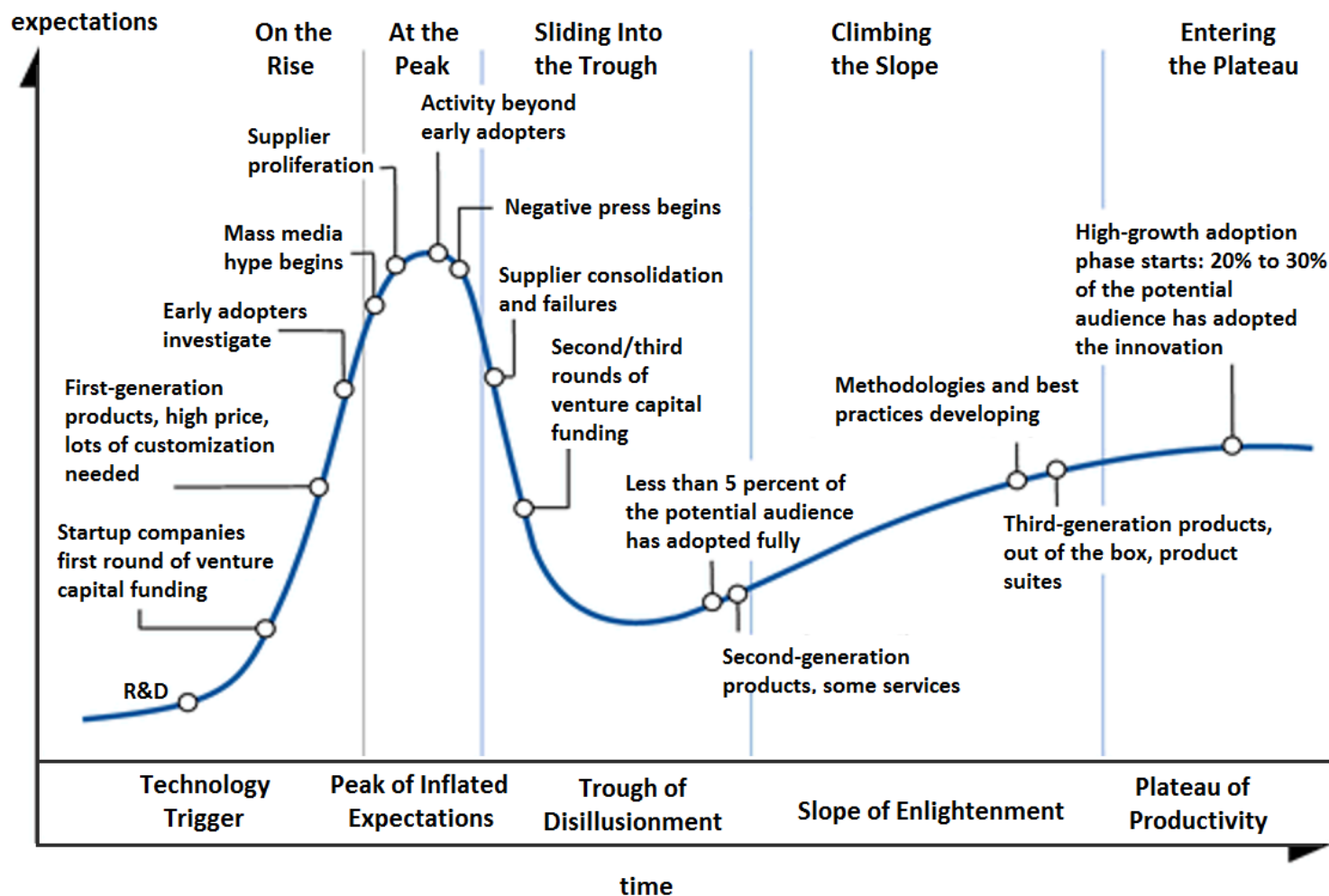
Technical architecture



Source: Wikimedia commons

Hype cycle

Proposed by Gartner, first published in 1995



Enterprise software

IT Software taxonomy

Enterprise Software

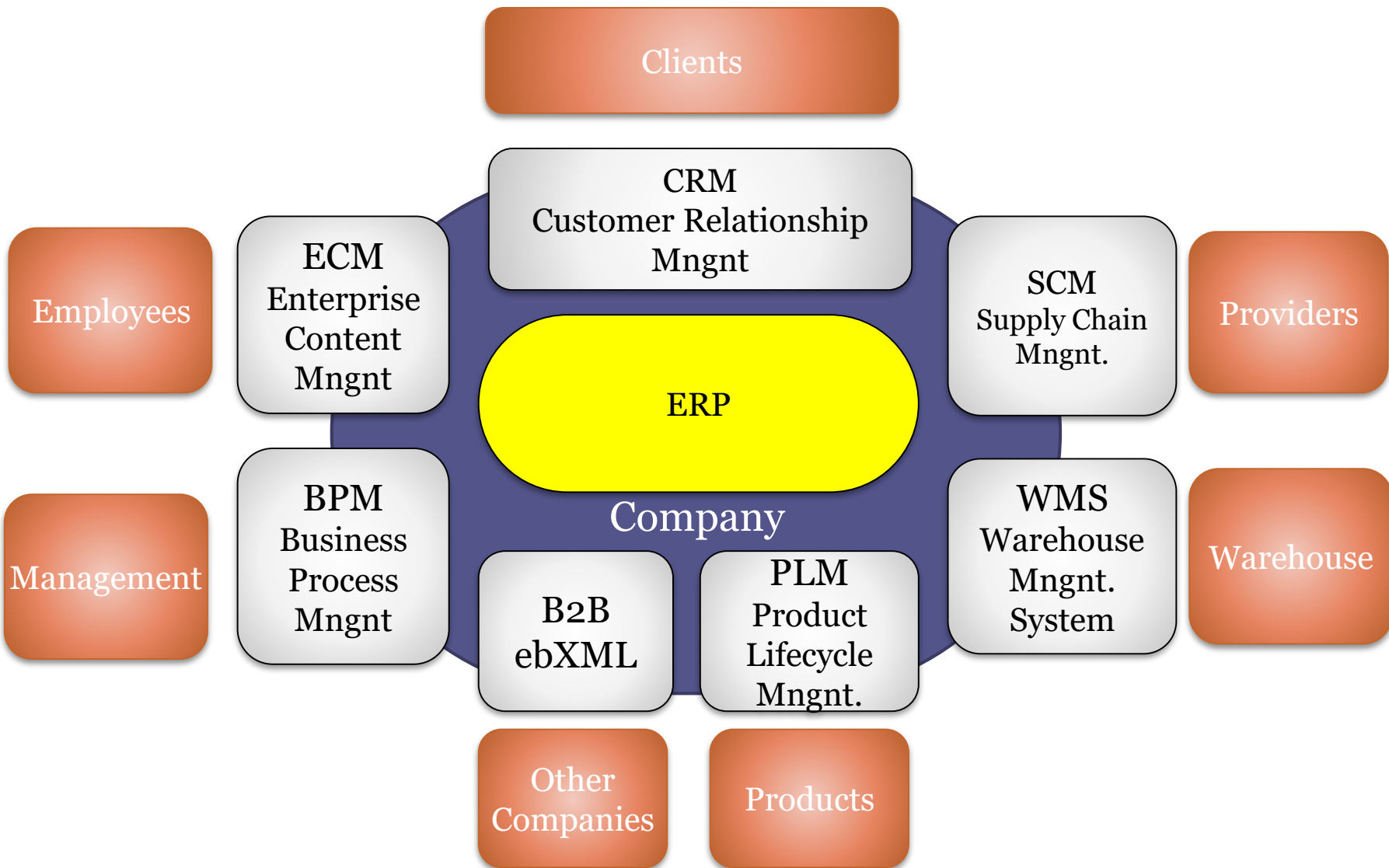
IT software taxonomy

System Classification:

CRM, ERP, SCM, ECM, PLM, EAI...



Enterprise information systems



ERP

ERP (Enterprise Resource Planning)

Appeared at the end 90s

2000 year increased its adoption

Enterprise Technological Structure

Central data base

Real time access

Centralized management of production, logistics, inventory, accounting, billing...

ERP

Advantages

Performance and quality

Reuse established
business best
practices

Process continuity

Information uniformity

Concept of unique and
single data

Continuous

technological update

Decision-making support

Challenges

Complexity:

Customization, Deployment,
Training, etc.

Involve people

Adapt existing processes

Overestimate software
benefits

Too much dependency on
some specific software

High costs to change to other
software

Adapting existing processes
to provided processes

ERP

Applications

SAP R/3

- 3 layers client/server architecture

- Based on a domain-specific language: ABAP

 - ABAP: Advanced Business Application Programming

- SAP = biggest european software company

Other systems:

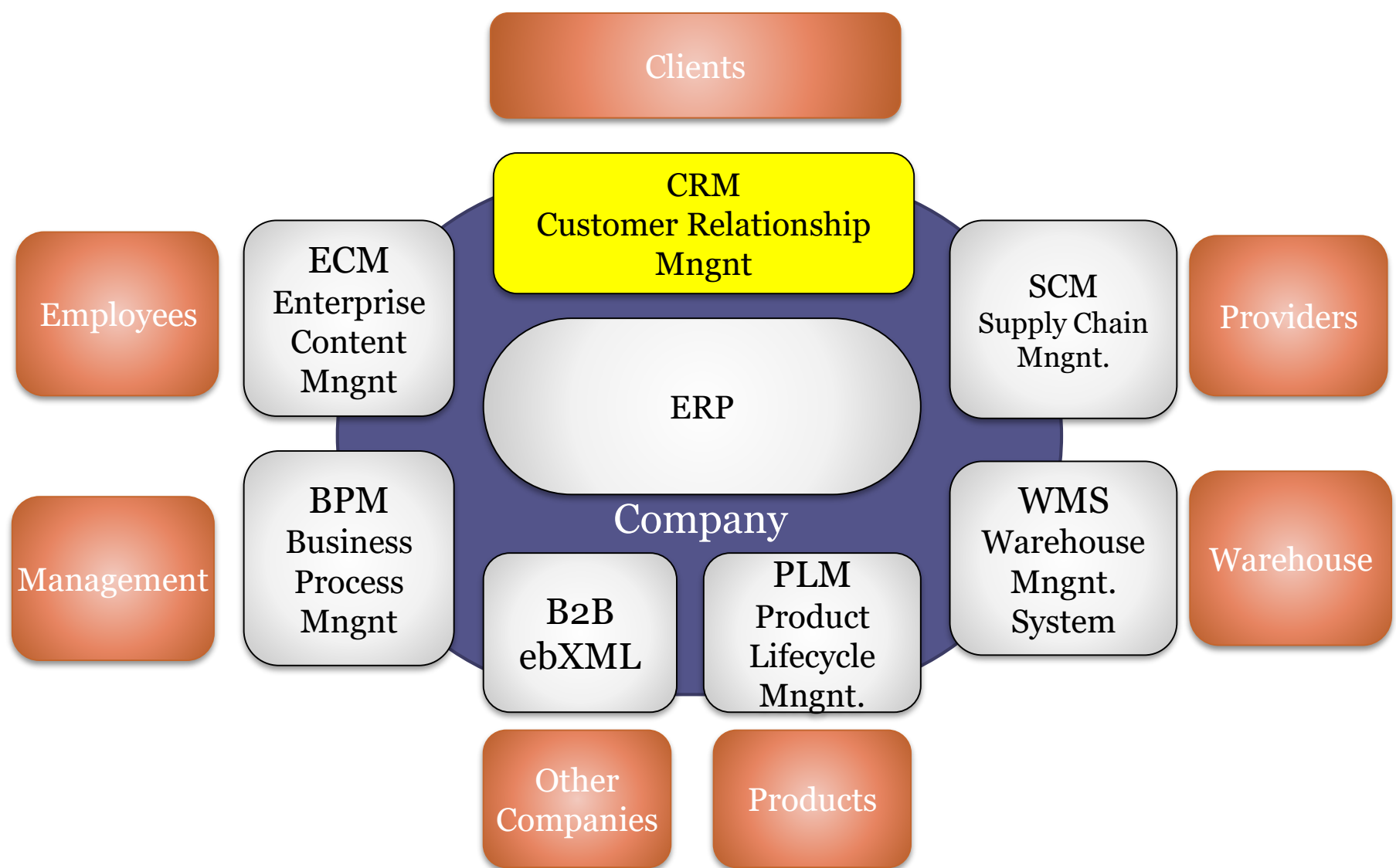
- Comercial:

 - Oracle Fusion, Microsoft Dynamics NAV, SAGE, ...

- Open source:

 - OpenERP, webERP, ...

Enterprise information systems



CRM - Customer Relationship Management

CRM - Customer Relationship Management

Manage relationships between Company and clients

Client lifecycle

Acquisition - Improve - Retain

Manages interactions with current and future clients

Involves:

Sales

Marketing

Client service

Call-centers

Technical support



CRM - Customer Relationship Management

Advantages

Helps identify best clients

Adapt products to client needs

Anticipate needs

Keeps track of client's contacts

Challenges

Client satisfaction

Labor cost reduction

Geographical and temporal diversity

Profile management and privacy

Social client and user communities

Combination with social networks (twitter, facebook,...)

Product reviews (Amazon, Booking,...)



CRM - Customer Relationship Management

Applications

Lots of CRM are integrated with ERP systems

SAP, Oracle, Microsoft CRM

Others:

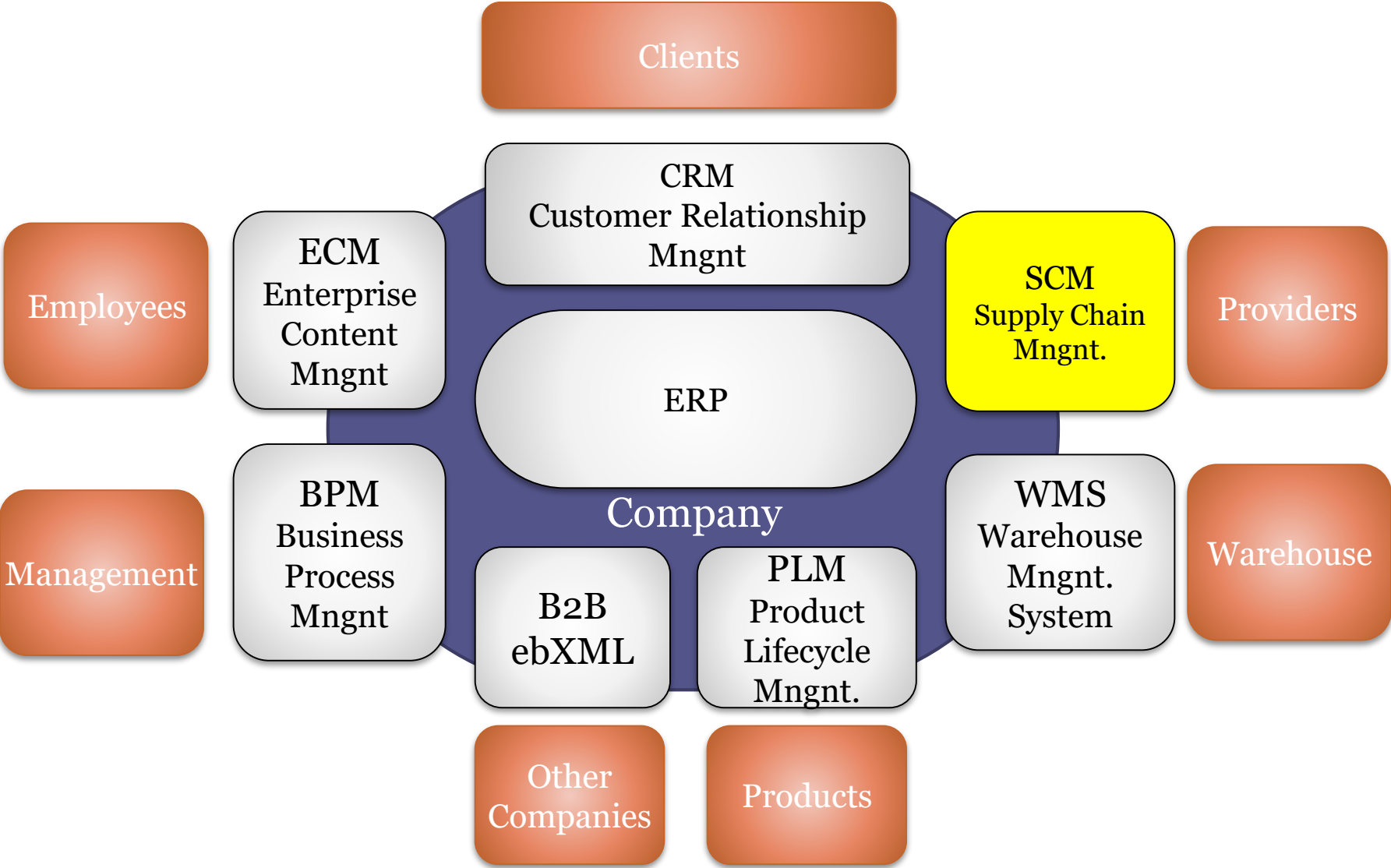
Salesforce.com

webCRM

...



Enterprise information systems



SCM - Supply Chain Management

SCM - Supply Chain Management

Processing client requirements

Purchase orders management

Inventory management

Goods reception and storage

Supplies and stocks management



SCM - Supply Chain Management

Advantages

Forecast future demands

Inventory control

Improve business relationships

Feedback and state of each element in supply chain

Challenges

Planning

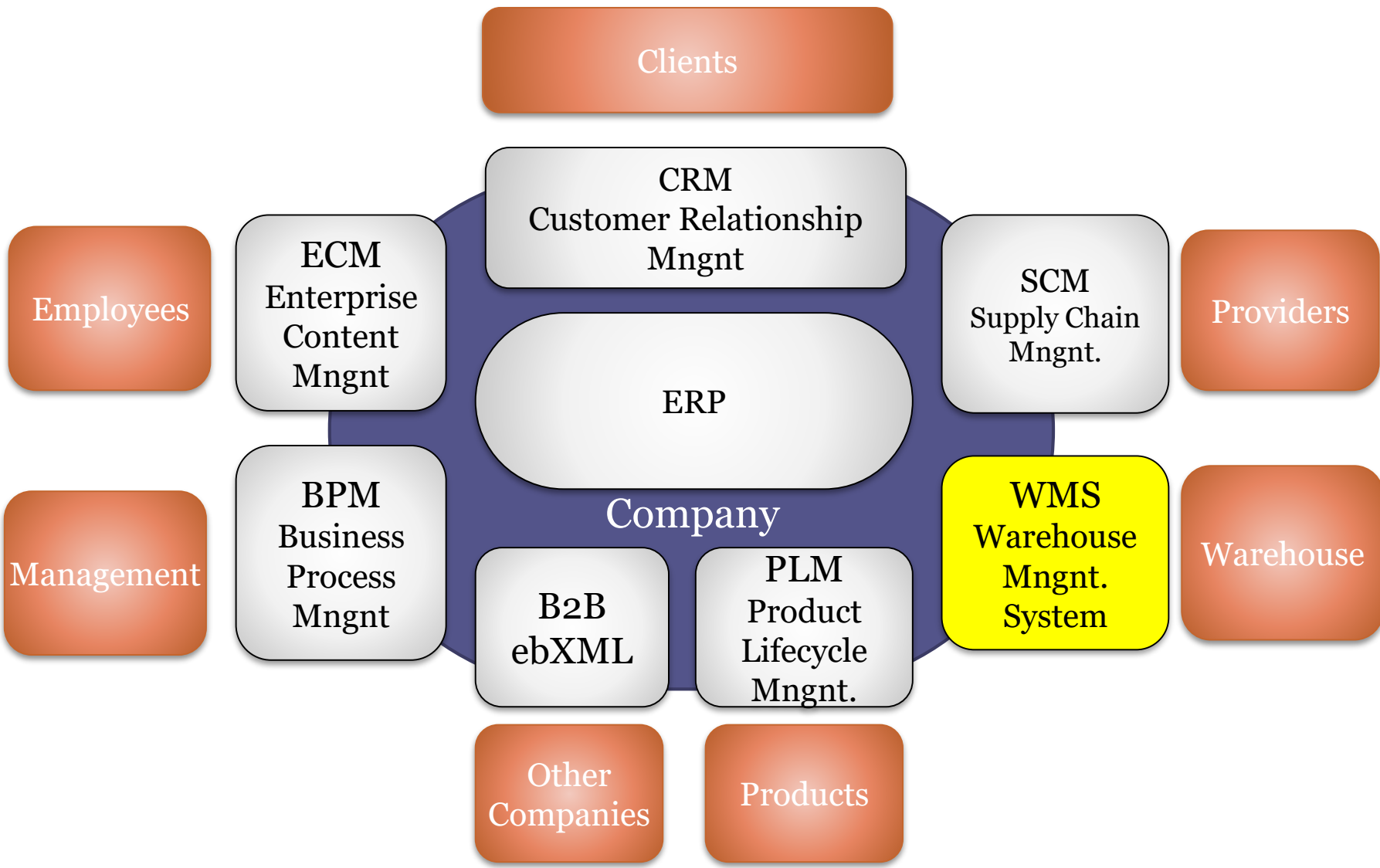
Lack of knowledge

Inaccurate forecasting demands

Lack of collaboration



Enterprise information systems



WMS - Warehouse Management Software

Warehouse Management Software

Product control in warehouses

Technology for identifying products

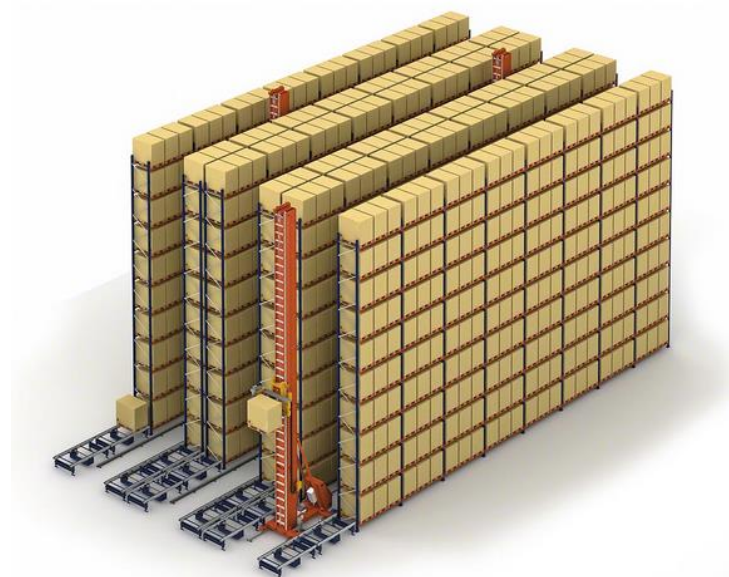
Picking, barcode scanners, RFID, etc.

Automated warehouses

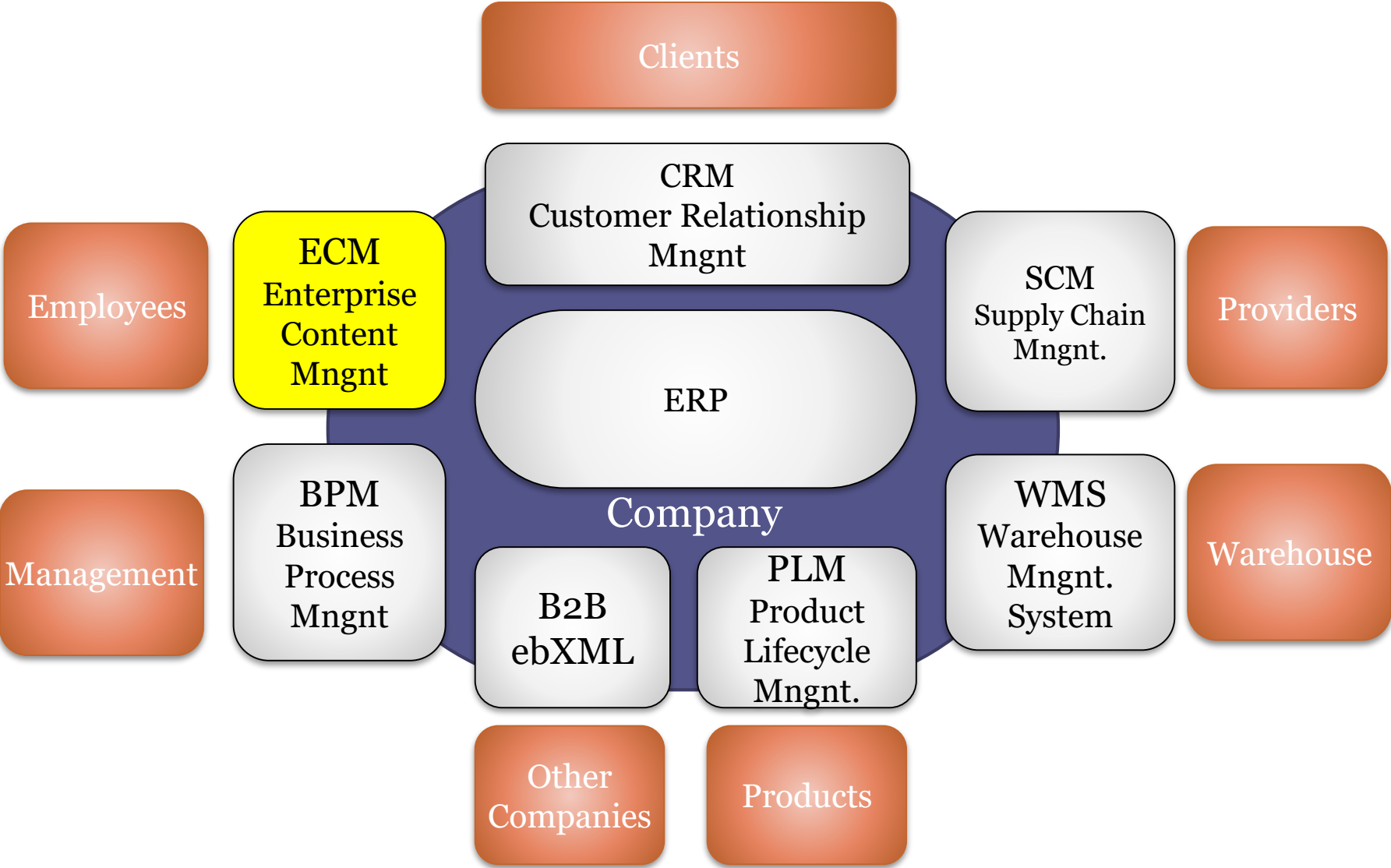
Stacker cranes, *miniloads*

Examples:

Mecalux EasyWMS



Enterprise information systems



ECM - Enterprise Content Management

Goal: Paperless office

Became popular in 2006

ECM - Enterprise Content Management

Components

Capture and recognition

- Scan and obtain documents

- Character recognition: OCR, HCR, etc.

Management:

- Document indexing

Storage

- Document management

Maintenance

- Security copies, archive, etc.

Sending

- Transformation and publishing

ECM - Enterprise Content Management

Systems

Microsoft Sharepoint

Oracle Content Management

EMC Documentum

Open source

Alfresco

LogicaDOC

Plone

...

ECM - Enterprise Content Management

Knowledge Management

Combines unstructured information sources

Groupware

Software for collaborative working groups

Example: Lotus Notes

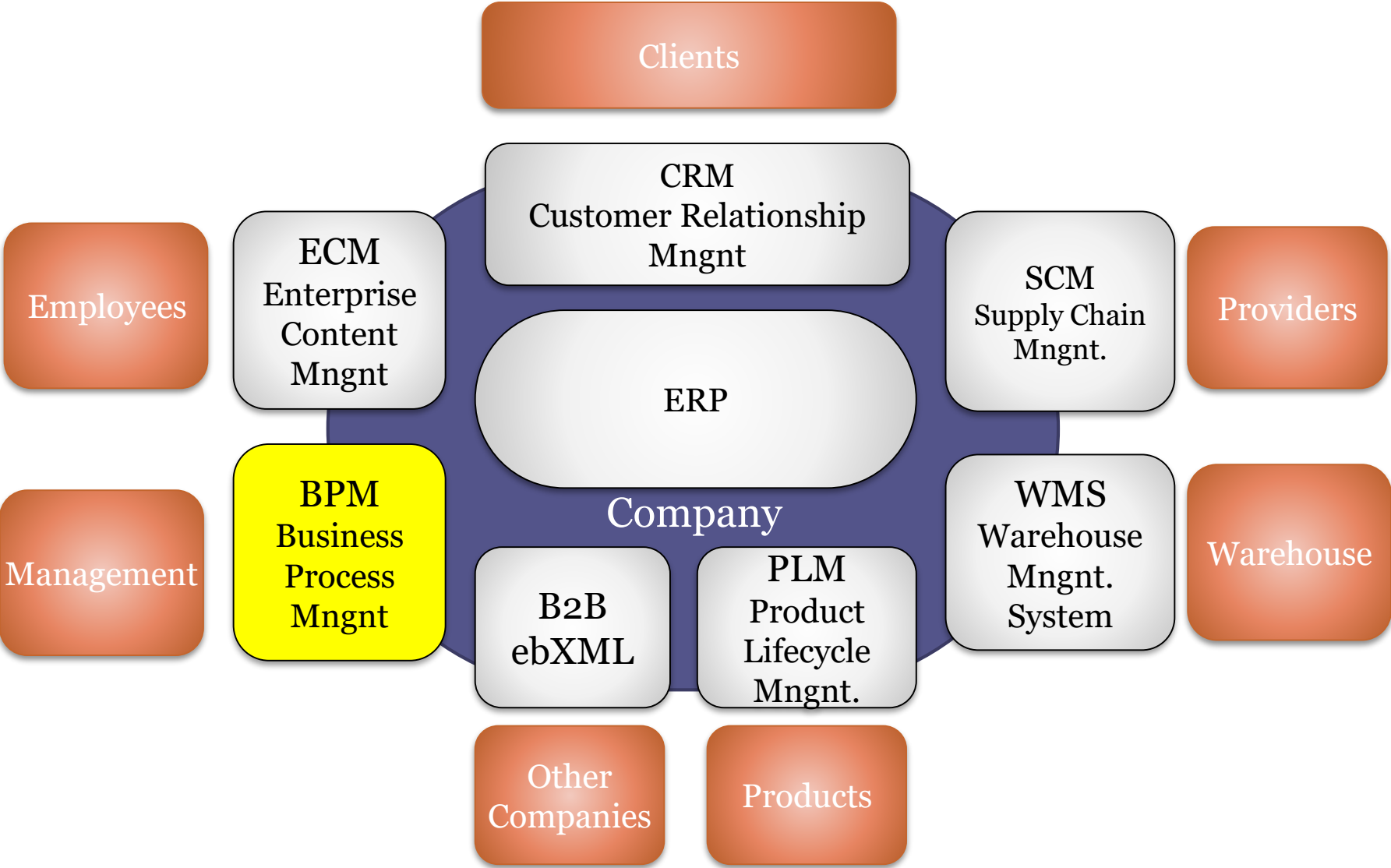
Enterprise Wikis

Example: Confluence

Document manager

Examples: LogicalDOC, Ricoh

Enterprise information systems



BPM - Business Process Management

Business process:

Set of procedures or activities with a business goal

Workflow:

Business workflow automation

BPM:

Business process lifecycle management through workflows

BPM - Business Process Management

Decision support systems

BI: Business Intelligence

Report creation

Data mining

Predictions and decision support

BPM - Business Process Management

OLAP (Online Analytical Processing) tools

Multidimensional analysis (OLAP Cube)

Enables to do operations

Sums, averages, etc. over groups of data

Solutions:

Microsoft, Oracle Business Objects,...

BPM - Business Process Management

Operational intelligence

Real-time monitoring

Balanced scorecard

Graphical visualizations of different metrics

Complex event processing

BPM - Business Process Management

Notations

BPEL (*Business Process Execution Language*)

- Defined by OASIS

- Defines relationships between web services (orchestration)

- Integration with WS-* standards

- More oriented to developers

BPMN (*Business Process Model and Notation*)

- Visual notation defined by OMG

- Represents business processes

- More oriented to business people

BPMN example

BPMN elements

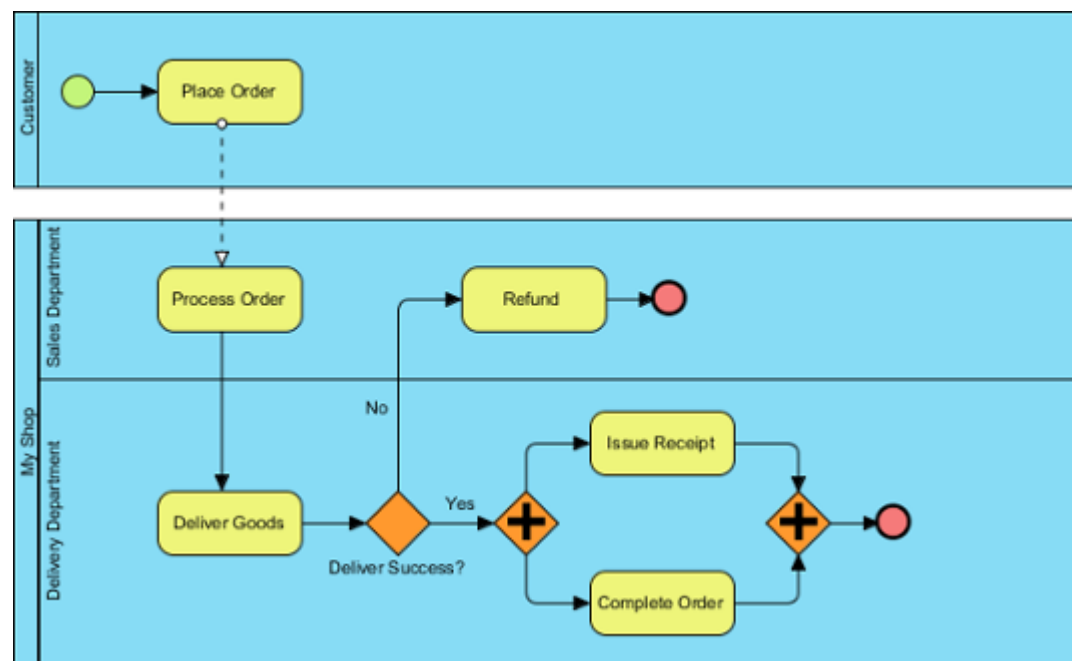
Events (start/intermediate/end)

Activities

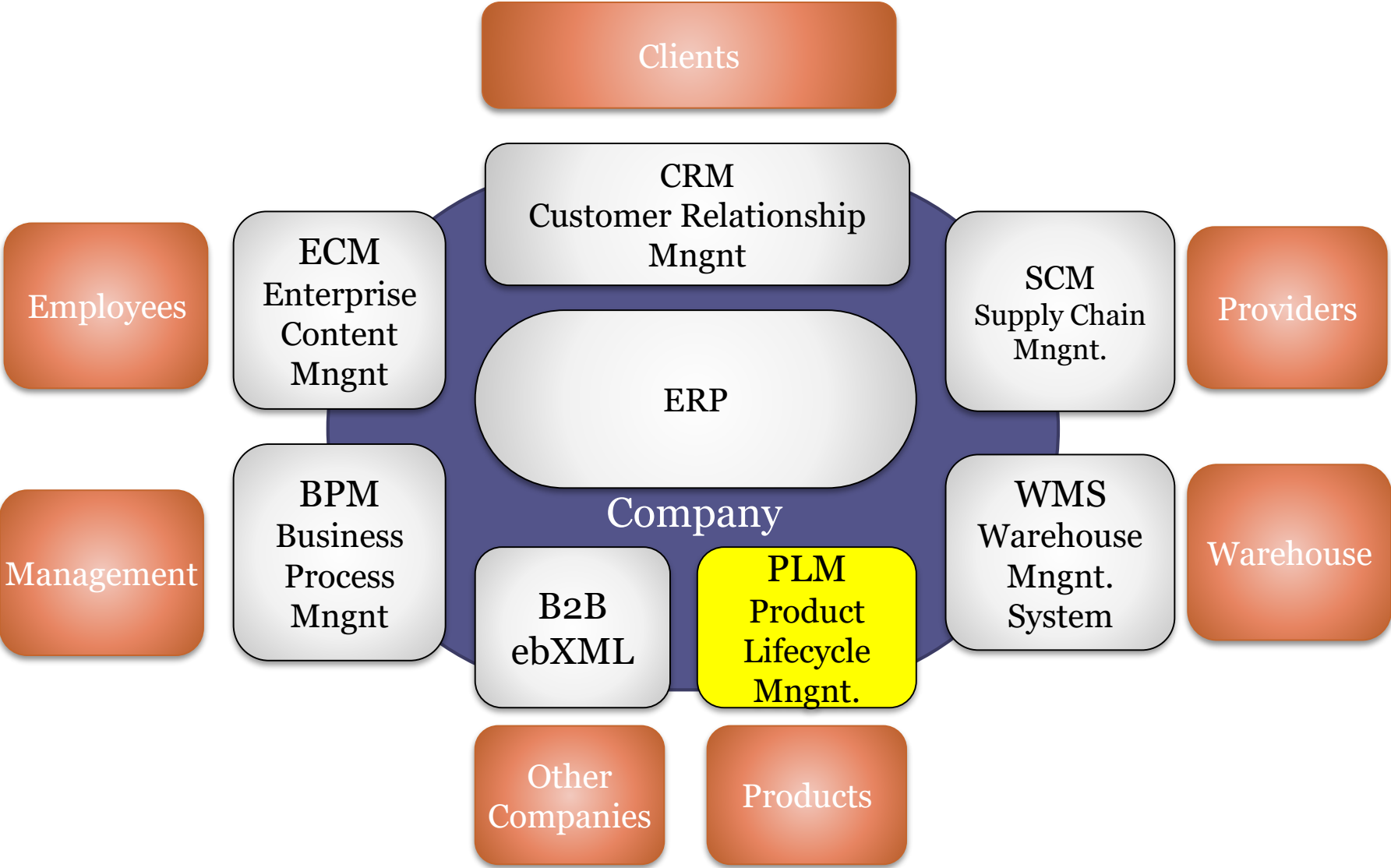
Gateways

Connections

Swim lanes



Enterprise information systems



PLM - Product Lifecycle Management

Integral product lifecycle management
Phases

Conceive

Design

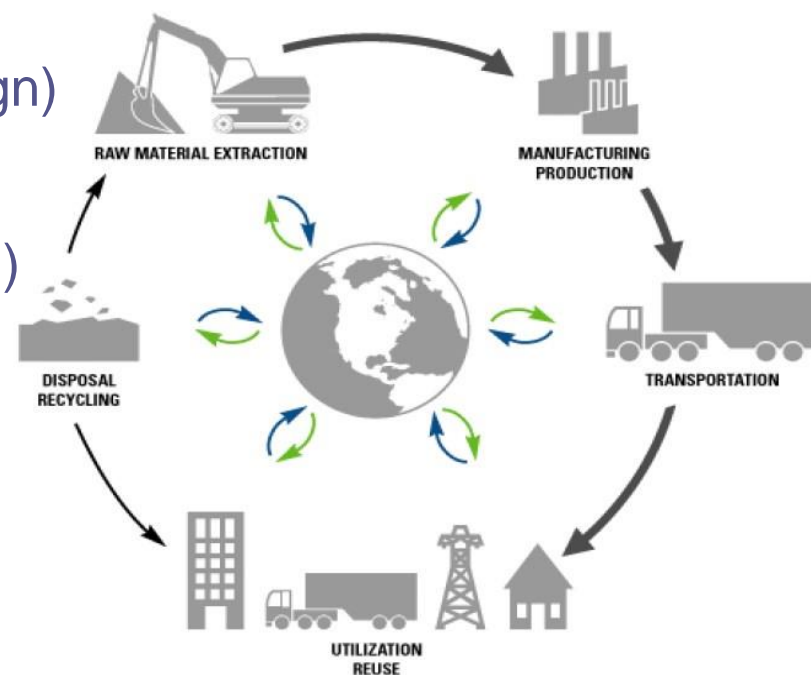
CAD tools (Computer Aided Design)

Realize

CAE (Computer Aided Engineering)

Service

Repair and maintenance



PLM - Product Lifecycle Management

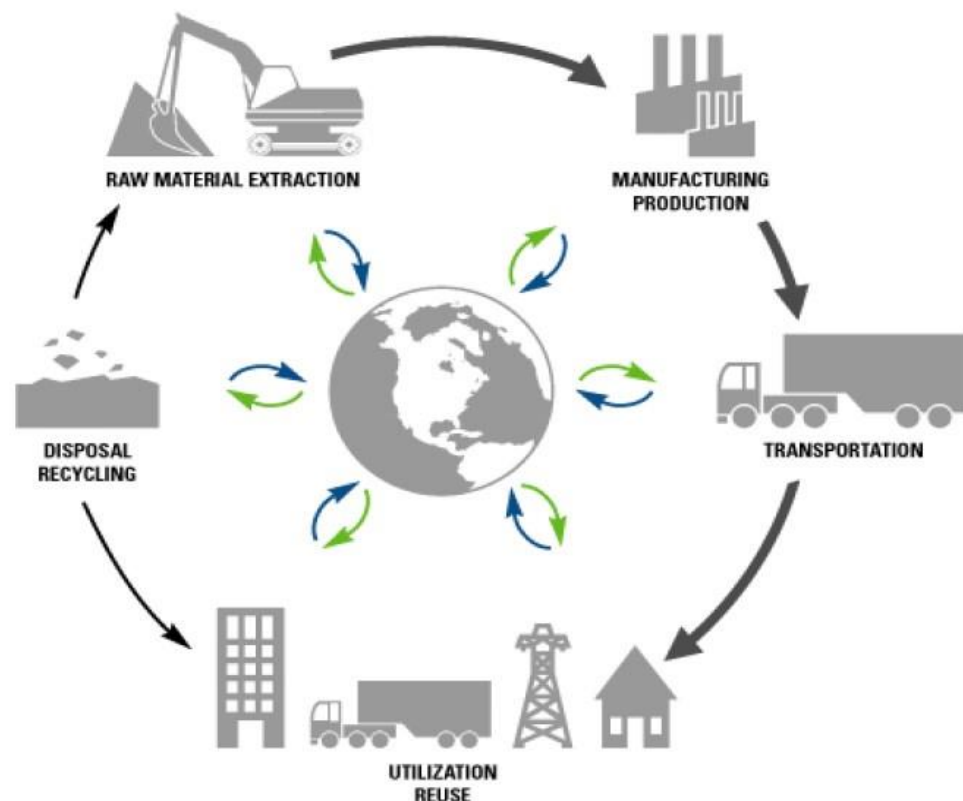
Solutions :

Siemens TeamCenter

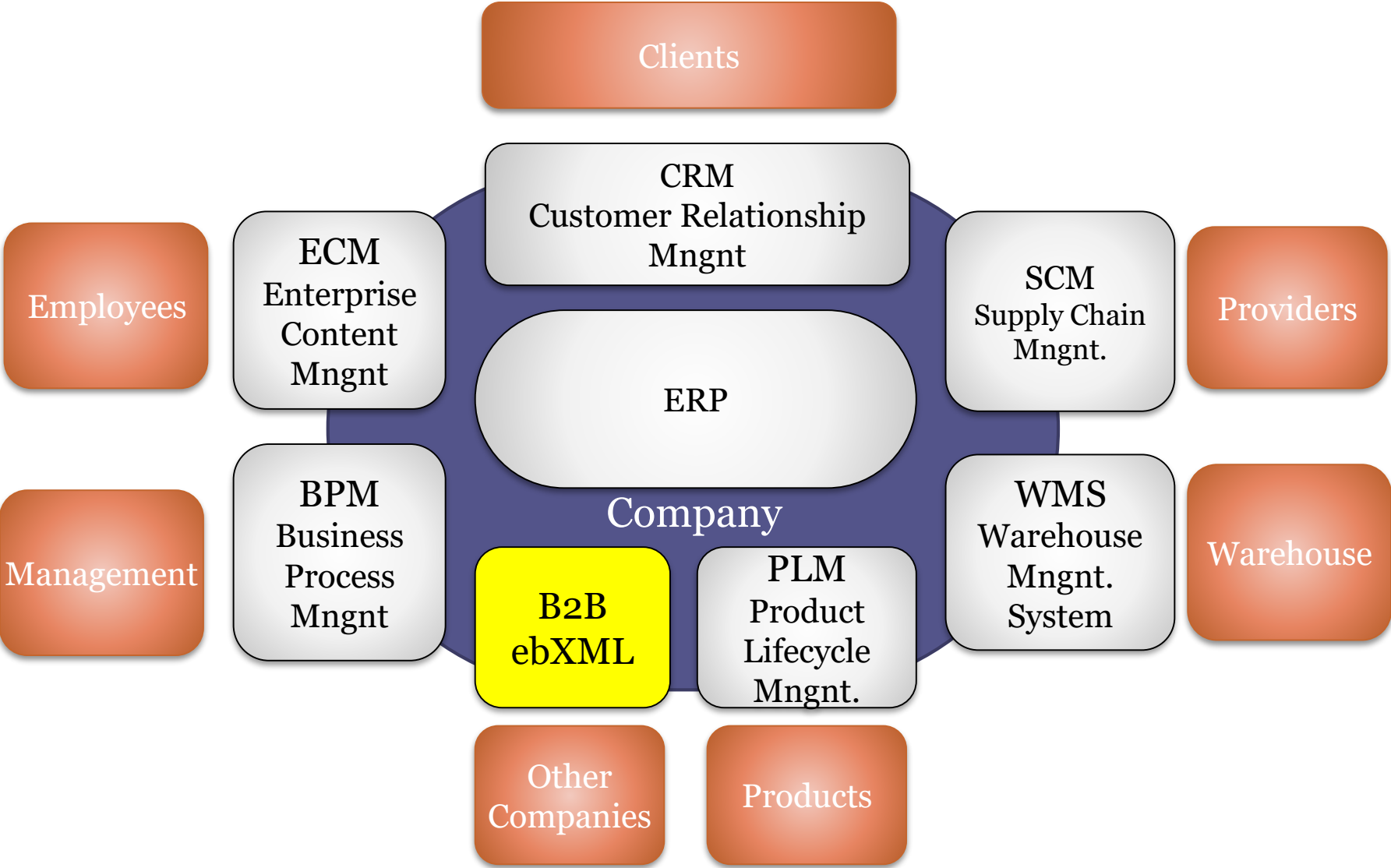
SAP PLM

Sopheon

...



Enterprise Information Systems



Relationship with external agents

Some common terms

B2B: Business-to-business

B2C: Business-to-consumer

C2C: Consumer-to-consumer

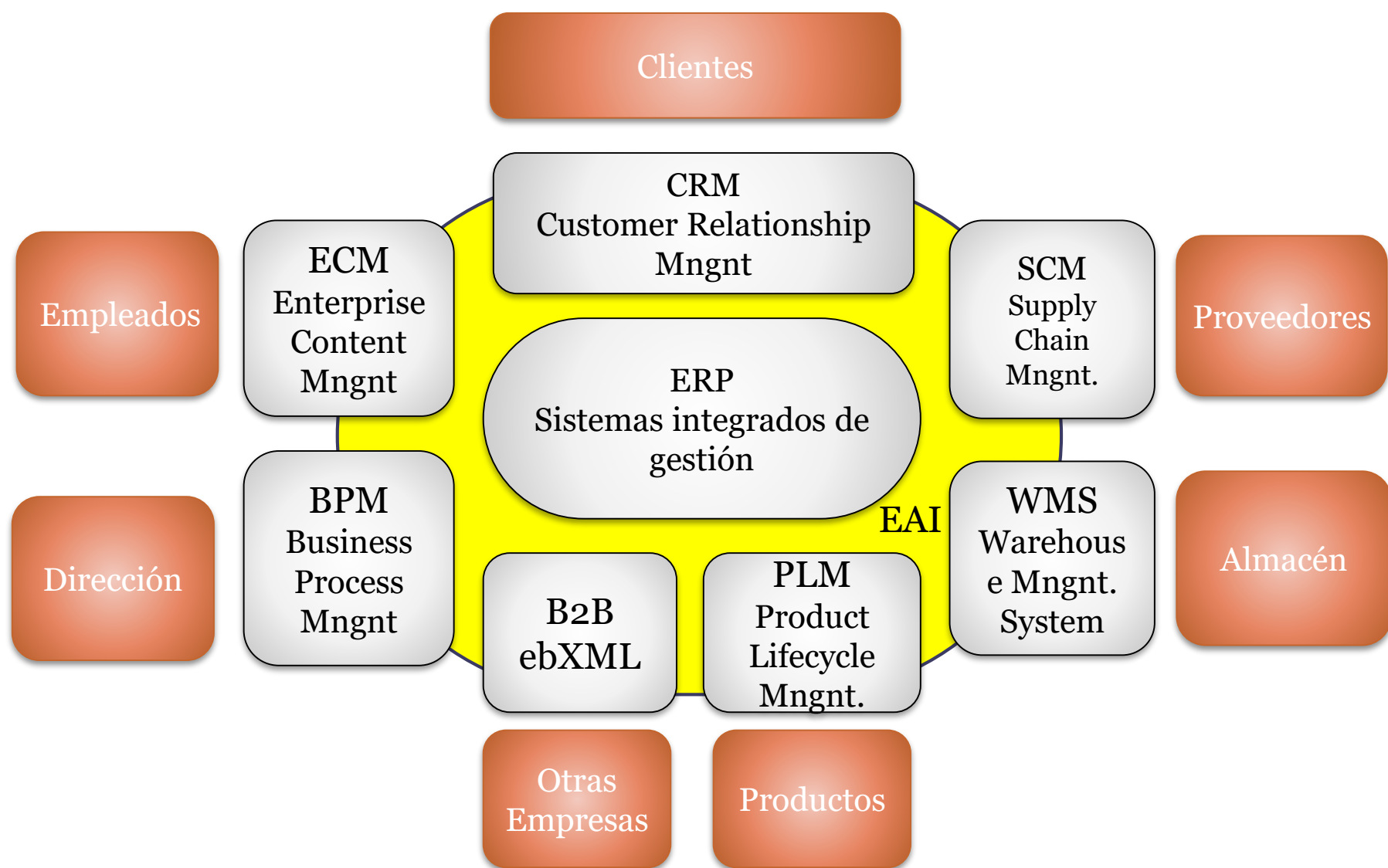
Most of e-commerce is B2B

Information exchange standards:

EDI (Electronic Data Interchange)

ebXML: XML based technologies for e-commerce

Enterprise Application Integration



EAI

EAI: Enterprise Application Integration

Glue between different systems

See:

Integration patterns (previous lesson)

Software product lines

Software product lines

Product line: products that share a set of functionalities to satisfy some given market segment

Goal:

- Reduce development effort

- Improve productivity

- Evolve from a single product to a product line

- Strategic reuse



Software product lines

Requirements

Identify generic solutions to common problems

Component based development

Generic Platforms

Software reuse

Generic architecture from which individual product architectures can be derived

Automatic system generation



Software and enterprise services

Towards services

Trend towards services

As a service family

Software as a service (SaaS)

Platform as a service (PaaS)

Infrastructure as a Service (IaaS)

Microservices

Service ecosystems

Service level terminology

Service Level Indicators (SLI)

Quantitative measure of some aspect of the level of service that is provided

Examples: error rate, system throughput, availability

Service Level Objective (SLO)

A target value or range of values for a service level that is measured by an SLI

Example: average search request latency should be less than 100 milliseconds

Service Level Agreement (SLA)

An explicit or implicit contract with your users that includes consequences of meeting (or missing) the SLOs they contain.

Service governance

Release management and deployment

Reliability and security

API management

Dependencies

Monitoring

Production support

Incidence response

On-call rotations

Cost model

Client onboarding

Documentation

Disaster recovery

Recommended books (free)
Site Reliability Engineering
<https://landing.google.com/sre/>

Software evolution

Software evolution

Timely updating the software

Reasons

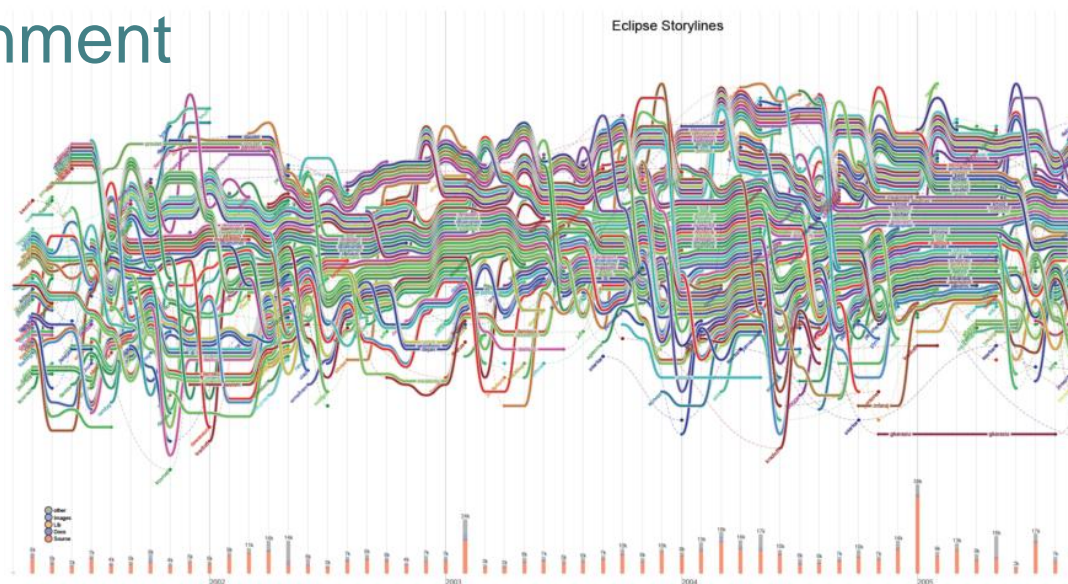
Changes in requisites with time

New functionalities or features

Changes in environment

Errors and bugs

Security risks



Lehman's laws of software evolution

Proposed in 1974 by Manny Lehman

Continuing Change

A system must be continually adapted or it becomes progressively less satisfactory

Increasing Complexity

As a system evolves, its complexity increases unless work is done to maintain or reduce it

Other laws from Lehman's:

Continuous growth

Conservation of familiarity

...



Manny Lehman
Fuente Wikidata



Software refactoring

Restructuring existing software without changing its external behaviour

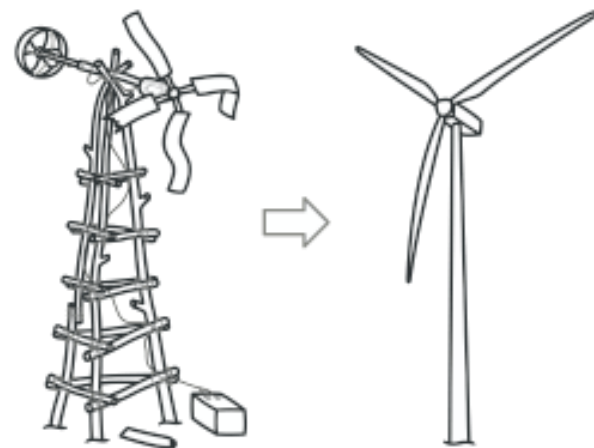
Keep functionality

Improve quality attributes

Goals

Avoid code smells

Pay technical debt



Legacy projects

Projects that are difficult to maintain or extend
Valuable software that you are afraid to change

Some reasons

Unfamiliarity

Developed by someone else/some time ago

No tests

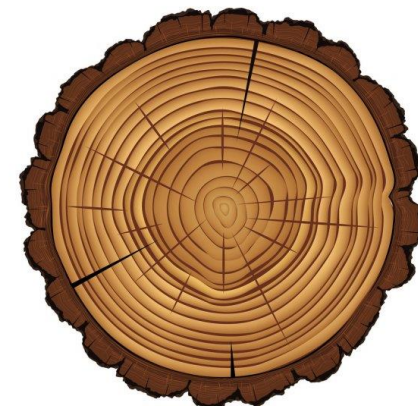
No documentation/outdated documentation

Reliance on external resources

Short deadlines

More info:

<https://understandlegacycode.com/>

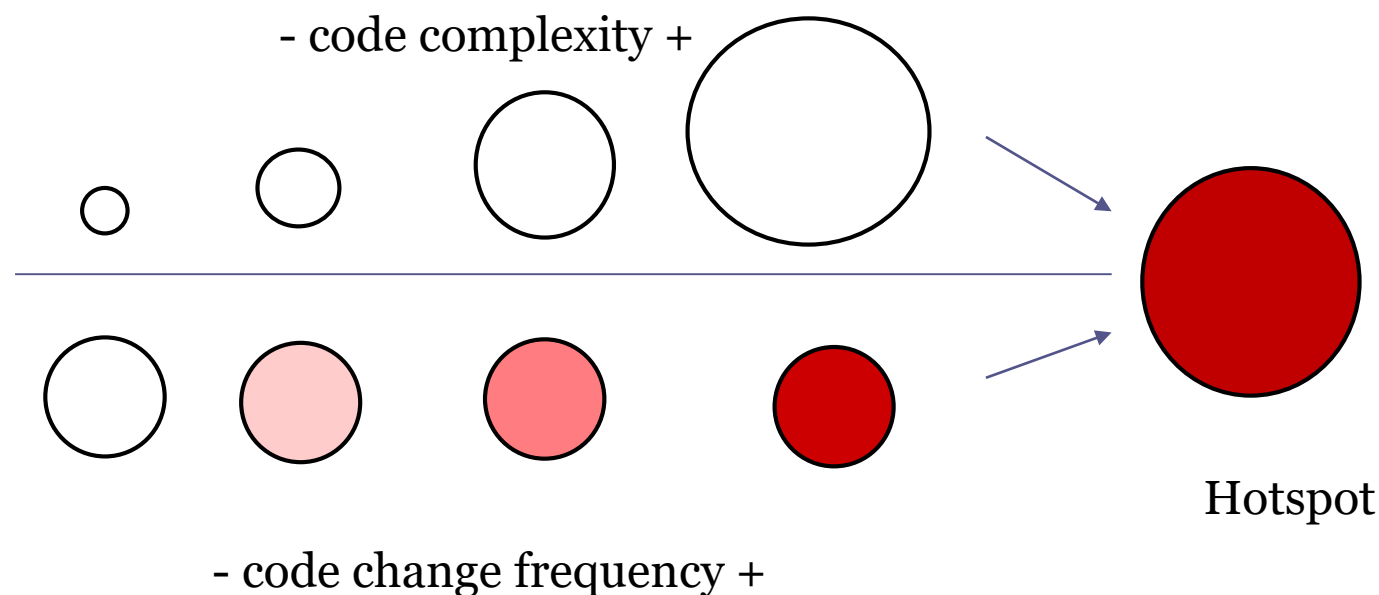


Behavioral code analysis

Identify system hotspots

Complex components that change frequently

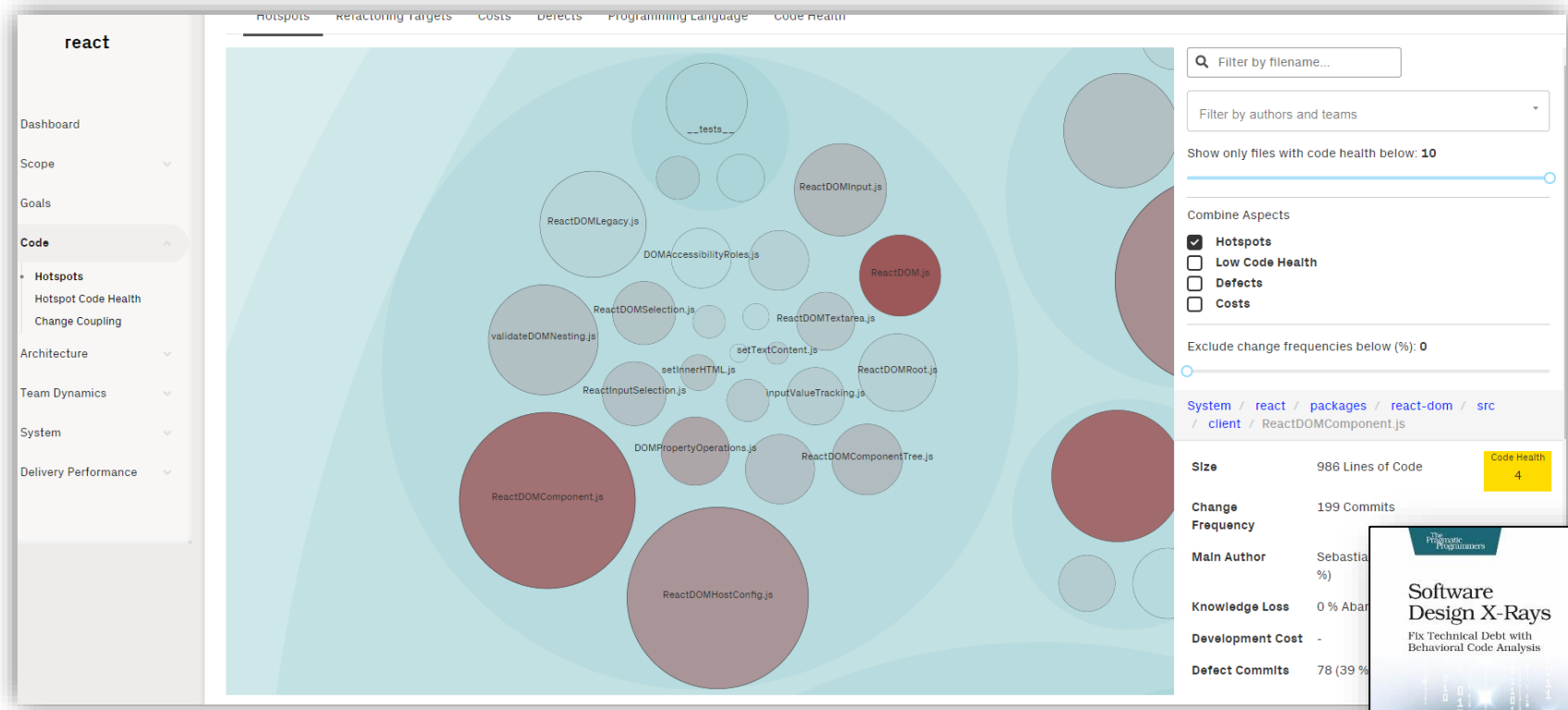
Good candidates for refactoring



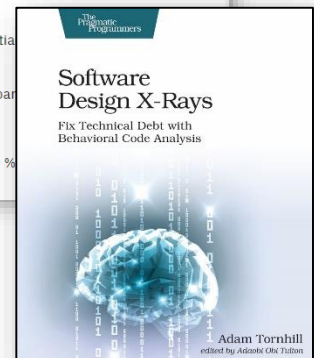
Behavioral code analysis

CodeScene tool: <https://codescene.io/>

Includes several tools for behavioral code analysis



More information: Book (Software design X-Rays, Adam Tornhill)



Evolutionary architectures

Incremental, guided change as a first principle

Main quality attribute: evolvability

Adoption of fitness function



End of presentation