Programutvecklingsprocessen

Utvecklingsprocess

- Processen att dela upp programvaruutvecklingsarbetet i olika faser för att förbättra design, produkthantering och projektledning.
- Metoden kan inkludera fördefinition av specifika leveranser och artefakter som skapas och slutförs av ett projektteam för att utveckla eller underhålla en applikation.
- De flesta moderna utvecklingsprocesser beskrivs som agila.
- https://en.wikipedia.org/wiki/Outline_of_software_development

Utvecklingsprocess

Kärnaktiviteter

- Requirements
- Design
- Engineering
- Construction
- Testing
- Debugging
- Deployment
- Maintenance

Utvecklingsprocess

Requirements

- In systems engineering and software engineering, requirements analysis focuses on the tasks that determine the needs or conditions to meet the new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements.

- Requirements analysis is critical to the success or failure of a systems or software project. The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

- https://en.wikipedia.org/wiki/Requirements_analysis

Utvecklingsprocess

Software design

- Software design is the process by which an agent creates a specification of a software artifact intended to accomplish goals, using a set of primitive components and subject to constraints.
- Software design may refer to either "all the activity involved in conceptualizing, framing, implementing, commissioning, and ultimately modifying complex systems" or "the activity following requirements specification and before programming, as ... [in] a stylized software engineering process."
- Software design usually involves problem-solving and planning a software solution. This includes both a low-level component and algorithm design and a high-level, architecture design.
- https://en.wikipedia.org/wiki/Software_design

Utvecklingsprocess

Software engineering

- Software engineering is the systematic application of engineering approaches to the development of software.
- https://en.wikipedia.org/wiki/Software_engineering

Utvecklingsprocess

Software construction

- Software construction is a software engineering discipline. It is the detailed creation of working meaningful software through a combination of coding, verification, unit testing, integration testing, and debugging. It is linked to all the other software engineering disciplines, most strongly to software design and software testing.

- https://en.wikipedia.org/wiki/Software_construction

Utvecklingsprocess

Software testing

- Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.
- Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.
- Test techniques include the process of executing a program or application with the intent of finding failures, and verifying that the software product is fit for use.
- https://en.wikipedia.org/wiki/Software_testing

Utvecklingsprocess

Debugging

- In computer programming and software development, debugging is the process of finding and resolving bugs (defects or problems that prevent correct operation) within computer programs, software, or systems.
- Debugging tactics can involve interactive debugging, control flow analysis, unit testing, integration testing, log file analysis, monitoring at the application or system level, memory dumps, and profiling.
- Many programming languages and software development tools also offer programs to aid in debugging, known as debuggers.
- https://en.wikipedia.org/wiki/Debugging

Utvecklingsprocess

Software deployment

- Software deployment is all of the activities that make a software system available for use.
- https://en.wikipedia.org/wiki/Software_deployment

Utvecklingsprocess

Software maintenance

- Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes.

- A common perception of maintenance is that it merely involves fixing defects. However, one study indicated that over 80% of maintenance effort is used for non-corrective actions.

- https://en.wikipedia.org/wiki/Software_maintenance

Utvecklingsprocess

Paradigmer och modeller

- Agile
- Cleanroom
- Incremental
- Prototyping
- Spiral
- V model
- Waterfall

Utvecklingsprocess

Agile software development

- In software development, agile (sometimes written Agile) practices involve discovering requirements and developing solutions through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s). It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change.

Utvecklingsprocess

Cleanroom software engineering

- The cleanroom software engineering process is a software development process intended to produce software with a certifiable level of reliability. The cleanroom process was originally developed by Harlan Mills and several of his colleagues including Alan Hevner at IBM.

- The focus of the cleanroom process is on defect prevention, rather than defect removal.

Utvecklingsprocess

Incremental build model

- The incremental build model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished.
- It involves both development and maintenance.
- The product is defined as finished when it satisfies all of its requirements.
- This model combines the elements of the waterfall model with the iterative philosophy of prototyping

Utvecklingsprocess

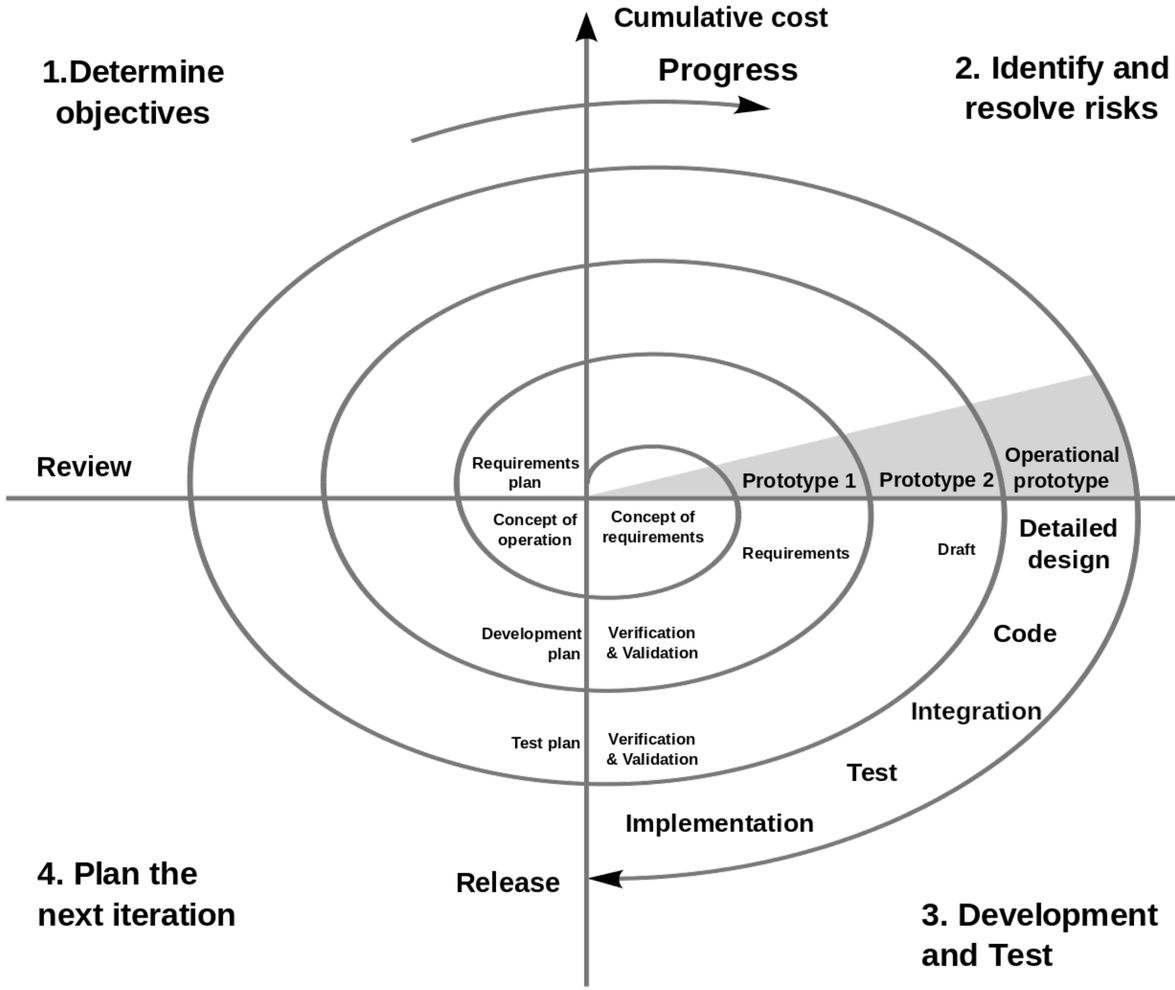
Software prototyping

- Software prototyping is the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed.
- It is an activity that can occur in software development and is comparable to prototyping as known from other fields, such as mechanical engineering or manufacturing.
- A prototype typically simulates only a few aspects of, and may be completely different from, the final product

Utvecklingsprocess

Spiral model

- The spiral model is a risk-driven software development process model.
- Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.



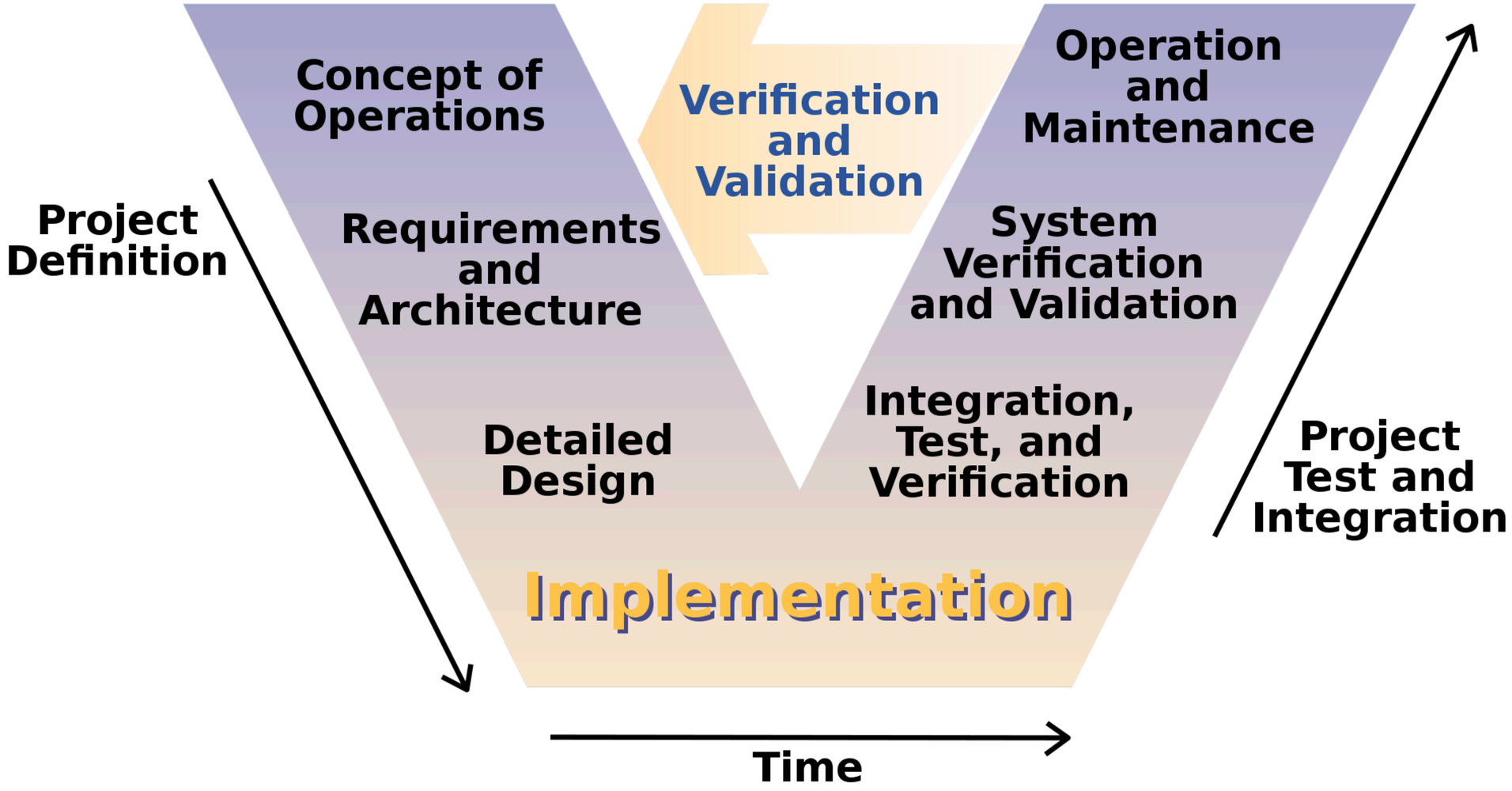
Utvecklingsprocess

V-Model

- In software development, the V-model represents a development process that may be considered an extension of the waterfall model, and is an example of the more general V-model.
- Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape.
- The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represents time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively.

Utvecklingsprocess

V-Model



Utvecklingsprocess

Waterfall model

- The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks.

- The approach is typical for certain areas of engineering design. In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, deployment and maintenance.

Utvecklingsprocess

Metoder och ramverk

- ASD
- DevOps
- DAD
- DSDM
- FDD
- IID
- Kanban
- Lean SD
- LeSS
- MDD
- MSF
- PSP
- RAD
- RUP
- SAFe
- Scrum
- SEMAT
- TSP
- OpenUP
- UP
- XP

Utvecklingsprocess

Feature-driven development

- Feature-driven development (FDD) is an iterative and incremental software development process.
- FDD blends a number of industry-recognized best practices into a cohesive whole.
- These practices are driven from a client-valued functionality (feature) perspective. Its main purpose is to deliver tangible, working software repeatedly in a timely manner in accordance with the Principles behind the Agile Manifesto.

Utvecklingsprocess

Kanban

- Kanban (Japanese; signboard or billboard) is a lean method to manage and improve work across human systems.
- This approach aims to manage work by balancing demands with available capacity, and by improving the handling of system-level bottlenecks.
- Work items are visualized to give participants a view of progress and process, from start to finish—usually via a Kanban board.
- Work is pulled as capacity permits, rather than work being pushed into the process when requested.

Utvecklingsprocess

Lean software development

- Lean software development is a translation of lean manufacturing principles and practices to the software development domain.
- Adapted from the Toyota Production System, it is emerging with the support of a pro-lean subculture within the Agile community.
- Lean offers a solid conceptual framework, values and principles, as well as good practices, derived from experience, that support agile organizations.

Utvecklingsprocess

Rapid application development

- Rapid-application development (RAD), also called rapid-application building (RAB), is both a general term for adaptive software development approaches, and the name for James Martin's method of rapid development.

- In general, RAD approaches to software development put less emphasis on planning and more emphasis on an adaptive process. Prototypes are often used in addition to or sometimes even instead of design specifications.

Utvecklingsprocess

Rational Unified Process

- The Rational Unified Process (RUP) is an iterative software development process framework created by the Rational Software Corporation.
- RUP is not a single concrete prescriptive process, but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs.
- RUP is a specific implementation of the Unified Process.

Utvecklingsprocess

Extreme programming

- Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.

- As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer requirements can be adopted.

Utvecklingsprocess

Supporting disciplines

- Configuration management
- Documentation
- Software quality assurance (SQA)
- Project management
- User experience

Utvecklingsprocess

Configuration management

- In software engineering, software configuration management (SCM or S/W CM) is the task of tracking and controlling changes in the software, part of the larger cross-disciplinary field of configuration management.
- SCM practices include revision control and the establishment of baselines.
  - If something goes wrong, SCM can determine what was changed and who changed it.
  - If a configuration is working well, SCM can determine how to replicate it across many hosts.

Utvecklingsprocess

Documentation

- Software documentation is written text or illustration that accompanies computer software or is embedded in the source code.
- The documentation either explains how the software operates or how to use it, and may mean different things to people in different roles.
- Documentation is an important part of software engineering. Types of documentation include:
    - Requirements – Statements that identify attributes, capabilities, characteristics, or qualities of a system. This is the foundation for what will be or has been implemented.
    - Architecture/Design – Overview of software. Includes relations to an environment and construction principles to be used in design of software components.
    - Technical – Documentation of code, algorithms, interfaces, and APIs.
    - End user – Manuals for the end-user, system administrators and support staff.
    - Marketing – How to market the product and analysis of the market demand.

Utvecklingsprocess

Software quality assurance (SQA / QA)

- Software quality assurance (SQA) is a means and practice of monitoring the software engineering processes and methods used in a project to ensure proper quality of the software.
- It may include ensuring conformance to standards or models, such as ISO/IEC 9126 (now superseded by ISO 25010), SPICE or CMMI.
- It includes standards and procedures that managers, administrators or even developers may use to review and audit software products and activities to verify that the software meets quality criteria which link to standards.
- SQA encompasses the entire software development process, including requirements engineering, software design, coding, code reviews, source code control, software configuration management, testing, release management and software integration. It is organized into goals, commitments, abilities, activities, measurements, verification and validation.

Utvecklingsprocess

Project management

- Software project management is an art and science of planning and leading software projects.
- It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled.

Utvecklingsprocess

User experience (UX)

- The user experience (UX or UE) is how a user interacts with and experiences a product, system or service.
- It includes a person's perceptions of utility, ease of use, and efficiency.
- Improving user experience is important to most companies, designers, and creators when creating and refining products because negative user experience can diminish use of the product and, therefore, profitability.
- User experience is subjective. However, the attributes that make up the user experience is objective.

Utvecklingsprocess

Practices

- ATDD
- BDD
- CCO
- CI
- CD
- DDD
- PP
- SBE
- Stand-up
- TDD

Utvecklingsprocess

Acceptance test–driven development

- Acceptance test–driven development (ATDD) is a development methodology based on communication between the business customers, the developers, and the testers.
- ATDD encompasses many of the same practices as specification by example (SBE), behavior-driven development (BDD), example-driven development (EDD), and support-driven development also called story test–driven development (SDD).
    - All these processes aid developers and testers in understanding the customer's needs prior to implementation and allow customers to be able to converse in their own domain language.

Utvecklingsprocess

Behavior-driven development

- In software engineering, behavior-driven development (BDD) is an Agile software development process that encourages collaboration among developers, QA and non-technical or business participants in a software project.
- It encourages teams to use conversation and concrete examples to formalize a shared understanding of how the application should behave.
- It emerged from test-driven development (TDD).
- Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design and object-oriented analysis and design to provide software development and management teams with shared tools and a shared process to collaborate on software development.

Utvecklingsprocess

Continuous integration

- In software engineering, continuous integration (CI) is the practice of merging all developers' working copies to a shared mainline several times a day.
- Grady Booch first proposed the term CI in his 1991 method, although he did not advocate integrating several times a day.
- Extreme programming (XP) adopted the concept of CI and did advocate integrating more than once per day – perhaps as many as tens of times per day.

Utvecklingsprocess

Continuous delivery

- Continuous delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time and, when releasing the software, doing so manually.
- It aims at building, testing, and releasing software with greater speed and frequency.
- The approach helps reduce the cost, time, and risk of delivering changes by allowing for more incremental updates to applications in production.
- A straightforward and repeatable deployment process is important for continuous delivery.

Utvecklingsprocess

Pair programming

- Pair programming is an agile software development technique in which two programmers work together at one workstation.
- One, the driver, writes code while the other, the observer or navigator, reviews each line of code as it is typed in.
- The two programmers switch roles frequently.
- While reviewing, the observer also considers the "strategic" direction of the work, coming up with ideas for improvements and likely future problems to address.
- This is intended to free the driver to focus all of their attention on the "tactical" aspects of completing the current task, using the observer as a safety net and guide.

Utvecklingsprocess

Test-driven development

- Test-driven development (TDD) is a software development process relying on software requirements being converted to test cases before software is fully developed, and tracking all software development by repeatedly testing the software against all test cases.
- This is opposed to software being developed first and test cases created later.
- Test-driven development is related to the test-first programming concepts of extreme programming, begun in 1999, but more recently has created more general interest in its own right.
- Programmers also apply the concept to improving and debugging legacy code developed with older techniques.

Utvecklingsprocess

Verktyg

- Compiler
- Debugger
- Profiler
- GUI designer
- Modeling
- IDE
- Build automation
- Release automation
- Infrastructure as code
- Testing

Utvecklingsprocess

Compiler

- In computing, a compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language).
- The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program.[

Utvecklingsprocess

Debugger

- A debugger or debugging tool is a computer program used to test and debug other programs (the "target" program).
- Typical debugging facilities include the ability to run or halt the target program at specific points, display the contents of memory, CPU registers or storage devices (such as disk drives), and modify memory or register contents in order to enter selected test data that might be a cause of faulty program execution.

Utvecklingsprocess

Profiling

- In software engineering, profiling is a form of dynamic program analysis that measures, for example, the space (memory) or time complexity of a program, the usage of particular instructions, or the frequency and duration of function calls.

- Most commonly, profiling information serves to aid program optimization, and more specifically, performance engineering.

- Profiling is achieved by instrumenting either the program source code or its binary executable form using a tool called a profiler (or code profiler).

- Profilers may use a number of different techniques, such as event-based, statistical, instrumented, and simulation methods.

Utvecklingsprocess

Graphical user interface builder

- A graphical user interface builder (or GUI builder), also known as GUI designer, is a software development tool that simplifies the creation of GUIs by allowing the designer to arrange graphical control elements (often called widgets) using a drag-and-drop WYSIWYG editor. (What You See Is What You Get.)

- Without a GUI builder, a GUI must be built by manually specifying each widget's parameters in source-code, with no visual feedback until the program is run.

Utvecklingsprocess

UML tool

- A UML tool is a software application that supports some or all of the notation and semantics associated with the Unified Modeling Language (UML), which is the industry standard general-purpose modeling language for software engineering.

Utvecklingsprocess

Integrated development environment

- An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.
- An IDE normally consists of at least a source code editor, build automation tools and a debugger.
- The boundary between an IDE and other parts of the broader software development environment is not well-defined; sometimes a version control system or various tools to simplify the construction of a graphical user interface (GUI) are integrated.
- Many modern IDEs also have a class browser, an object browser, and a class hierarchy diagram for use in object-oriented software development.

Utvecklingsprocess

Build automation

- Build automation is the process of automating the creation of a software build and the associated processes including: compiling computer source code into binary code, packaging binary code, and running automated tests.

Utvecklingsprocess

Standards and Bodies of Knowledge

- BABOK
- CMMI
- IEEE standards
- ISO 9001
- ISO/IEC standards
- PMBOK
- SWEBOK
- ITIL
- IREB

Utvecklingsprocess

ISO 9000

- The ISO 9000 family of quality management systems (QMS) is a set of standards that helps organizations ensure they meet customer and other stakeholder needs within statutory and regulatory requirements related to a product or service.
- ISO 9000 deals with the fundamentals of QMS, including the seven quality management principles that underlie the family of standards.
- ISO 9001 deals with the requirements that organizations wishing to meet the standard must fullfil.

Utvecklingsprocess

ITIL

- ITIL (formerly Information Technology Infrastructure Library) is a set of detailed practices for IT service management (ITSM) that focuses on aligning IT services with the needs of business.

UML-diagram

Unified Modeling Language

- UML ger systemutvecklare, programvarukonstruktörer, testingenjörer och kunder en visuell representation av det system som skall utvecklas. UML är vanligt vid objektorienterad programmering, och flera utvecklingsverktyg kan generera källkodsstubbar och/eller körbar kod direkt ur UML-modellen.

UML-diagram

Unified Modeling Language

- UML definierar notation/diagram för modeller användbara för kravmodellering, analys, design och implementation av IT/datasystem:
- **Användningsfallsmodeller**, som visar användare av systemet (så kallade aktörer) och funktionalitet i form av användningsfall.
- **Klassmodeller** som presenterar begrepp (klasser) och deras inbördes relationer. Klasser kapslar in attribut (datavärden) och tillhandahåller metoder som kan anropas. Klasser kan grupperas i paket för att öka översiktligheten.
- **Interaktionsdiagram** som visar hur objekt, som är förekomster av klasser, samverkar dynamiskt. Detta visas med hjälp av sekvensdiagram (som har svenskt ursprung inom Ericsson-sfären) eller kommunikationsdiagram. Ett kommunikationsdiagram kan användas för att visa hur man implementerar ett användningsfall.
- **Tillståndsdiagram** som visar hur en tillståndsmaskin arbetar, dvs vilka tillstånd och tillståndsövergångar som är tillåtna.
- **Aktivitetsdiagram** som kan visa flöden av aktiviteter, till exempel hur en verksamhets- eller tillverkningsprocess utförs. Aktivitetsdiagram är en kraftfull mekanism för att komplettera användningsfallsbeskrivningar.
- **Deployment diagram** som visar vilka noder som finns i systemet eller i ett nätverk av system och vilka komponenter som finns allokerade till varje nod. Diagrammet visar även vilka noder som samarbetar och vilka protokoll som används.

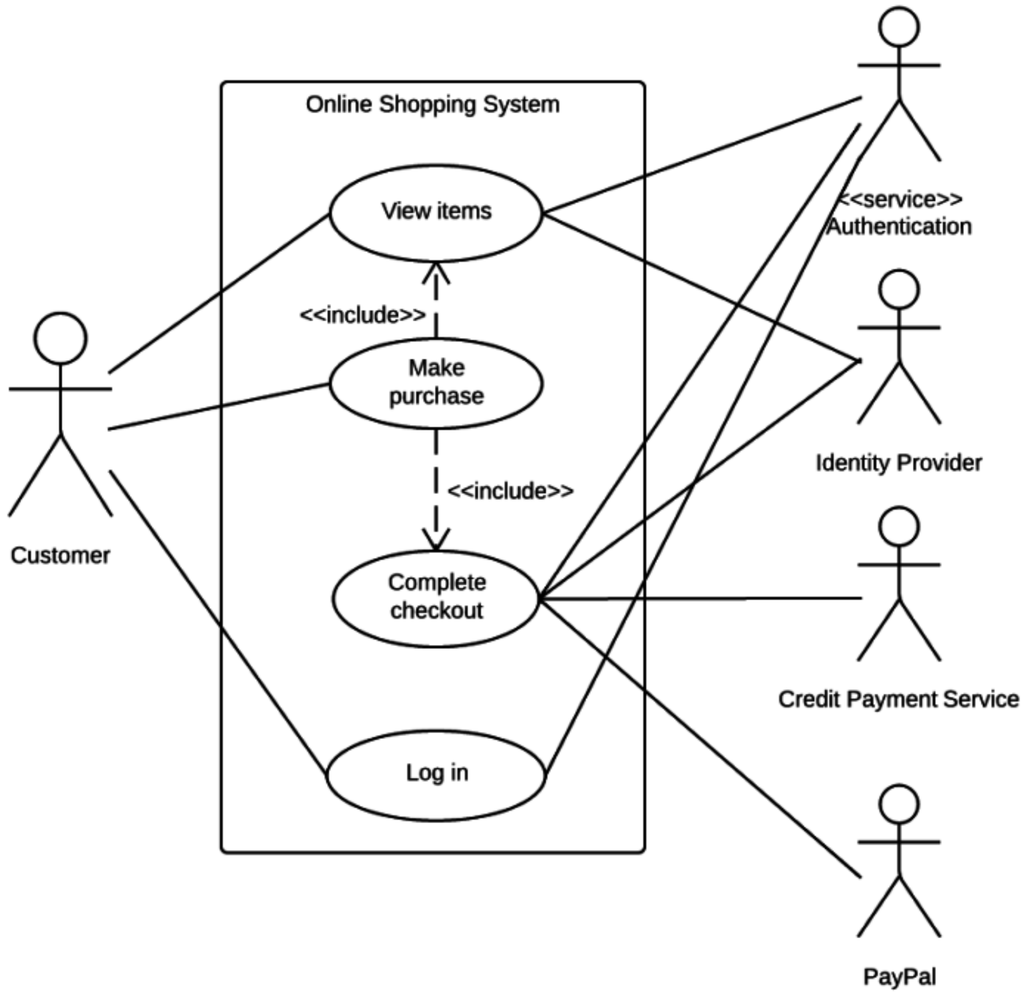
UML-diagram

Användningsfall

- Ett användningsfall inom programvaruutveckling är ett sätt att inhämta krav på ett nytt system eller ändring på befintlig programvara.

- Varje användningsfall innehåller ett eller flera scenarier som beskriver hur systemet ska interagera med sin omvärld (slutanvändaren eller annat system) för att uppfylla ett specifikt affärsmål.

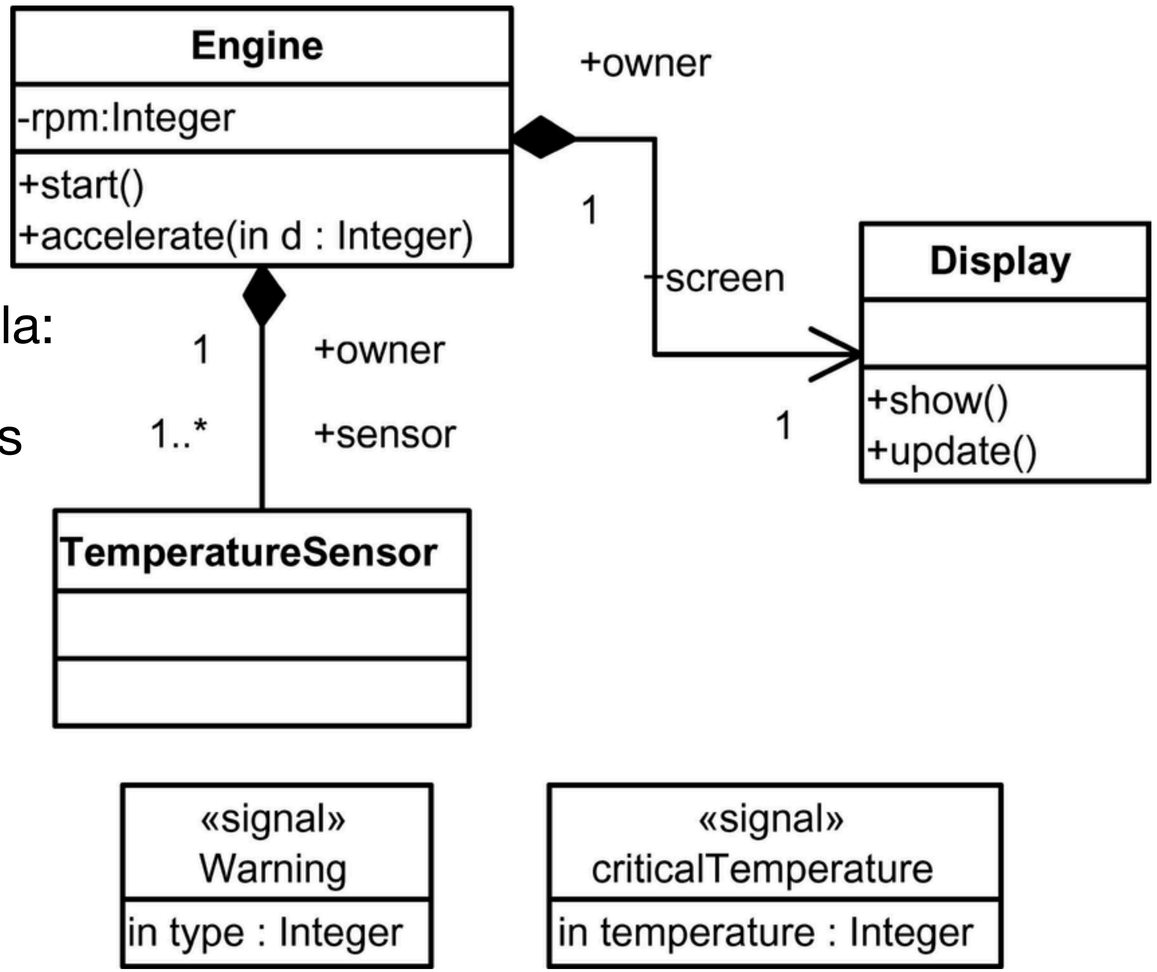
- https://www.lucidchart.com/pages/uml-use-case-diagram



UML-diagram

Klassmodeller

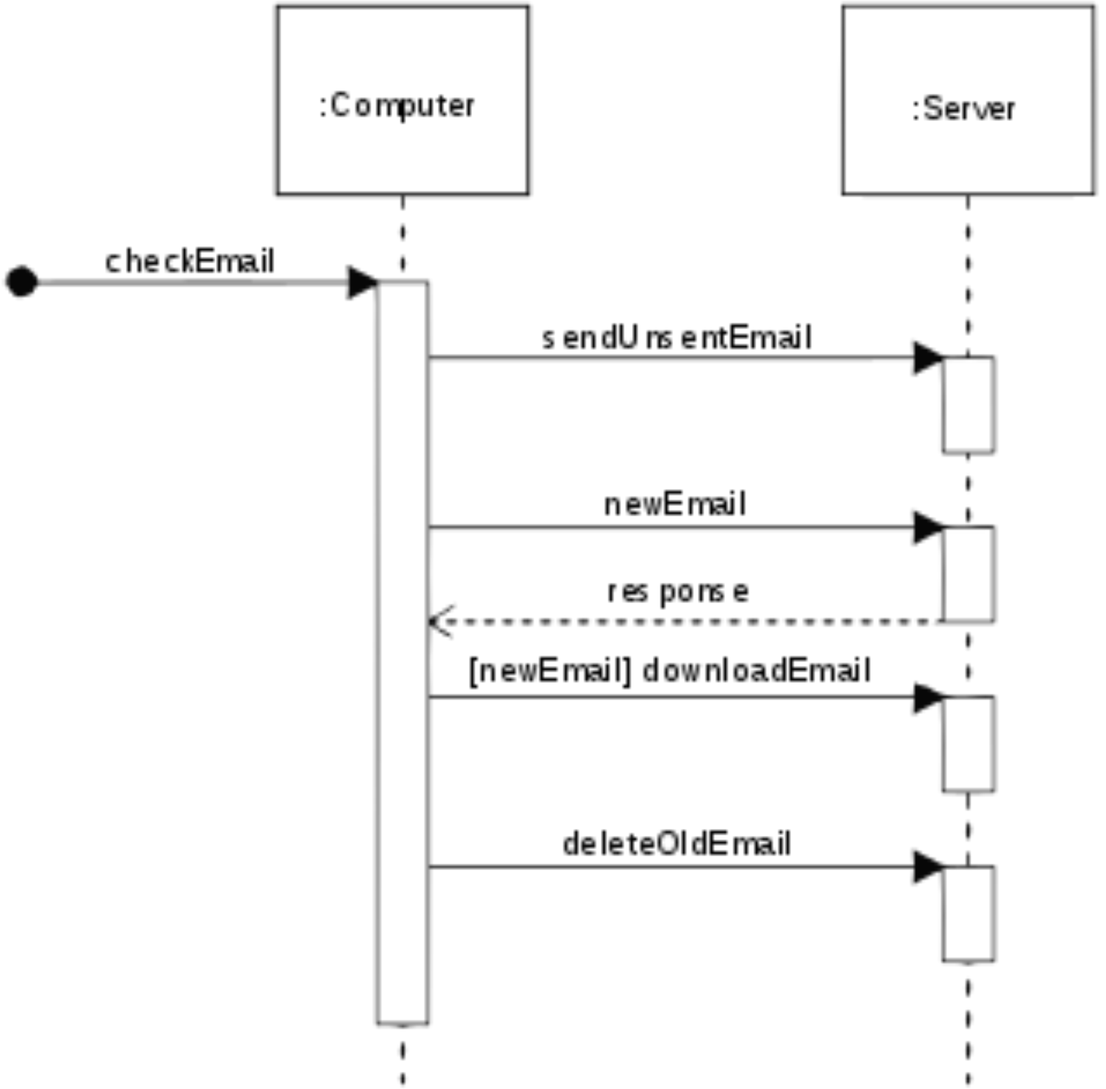
- Klassmodeller, klassdiagram i UML. Visar det centrala: strukturen i ett programvarusystem. Vad handlar systemet om, vilka viktiga verksamhetsbegrepp finns det, hur är de relaterade inbördes.
  - För kravställaren, beställaren, domänexperten: närmast en OMG-standardiserad begreppskarta eller mind map.
  - För programmeraren: ett sätt att efter kompletteringar automatiskt generera ut sina klassdefinitioner i objektorienterad programmering och sin ev. datamodell och databasdefinitioner, till exempel i SQL:s Data Definition Language.



UML-diagram

Interaktionsdiagram

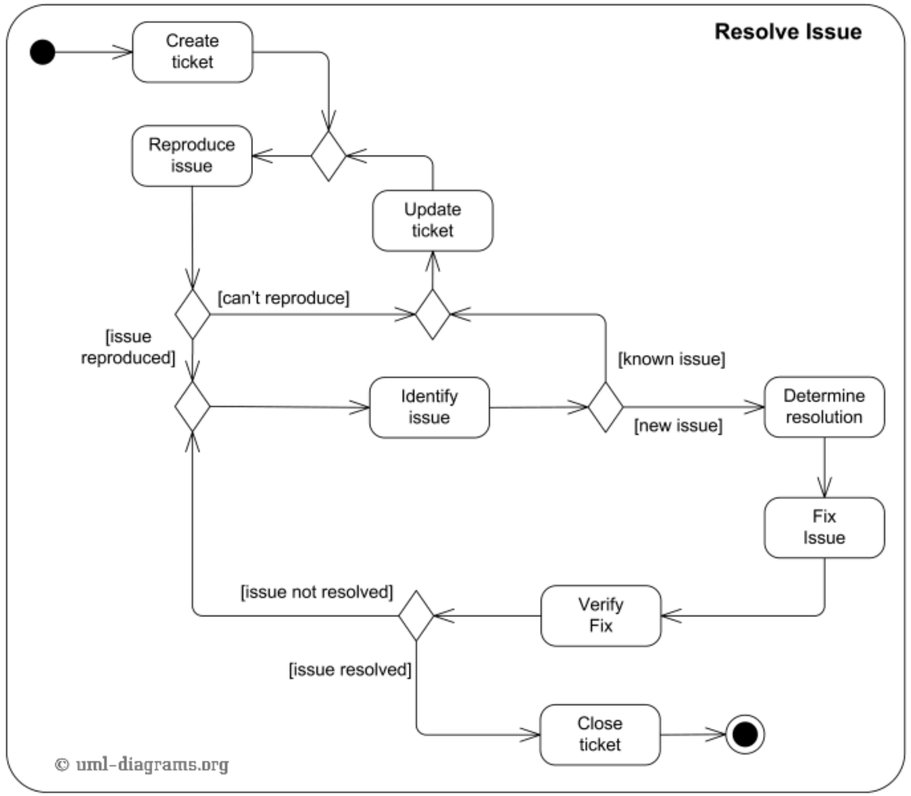
- From the term Interaction, it is clear that the diagram is used to describe some type of interactions among the different elements in the model. This interaction is a part of dynamic behavior of the system.
- The purpose of interaction diagrams is to visualize the interactive behavior of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.

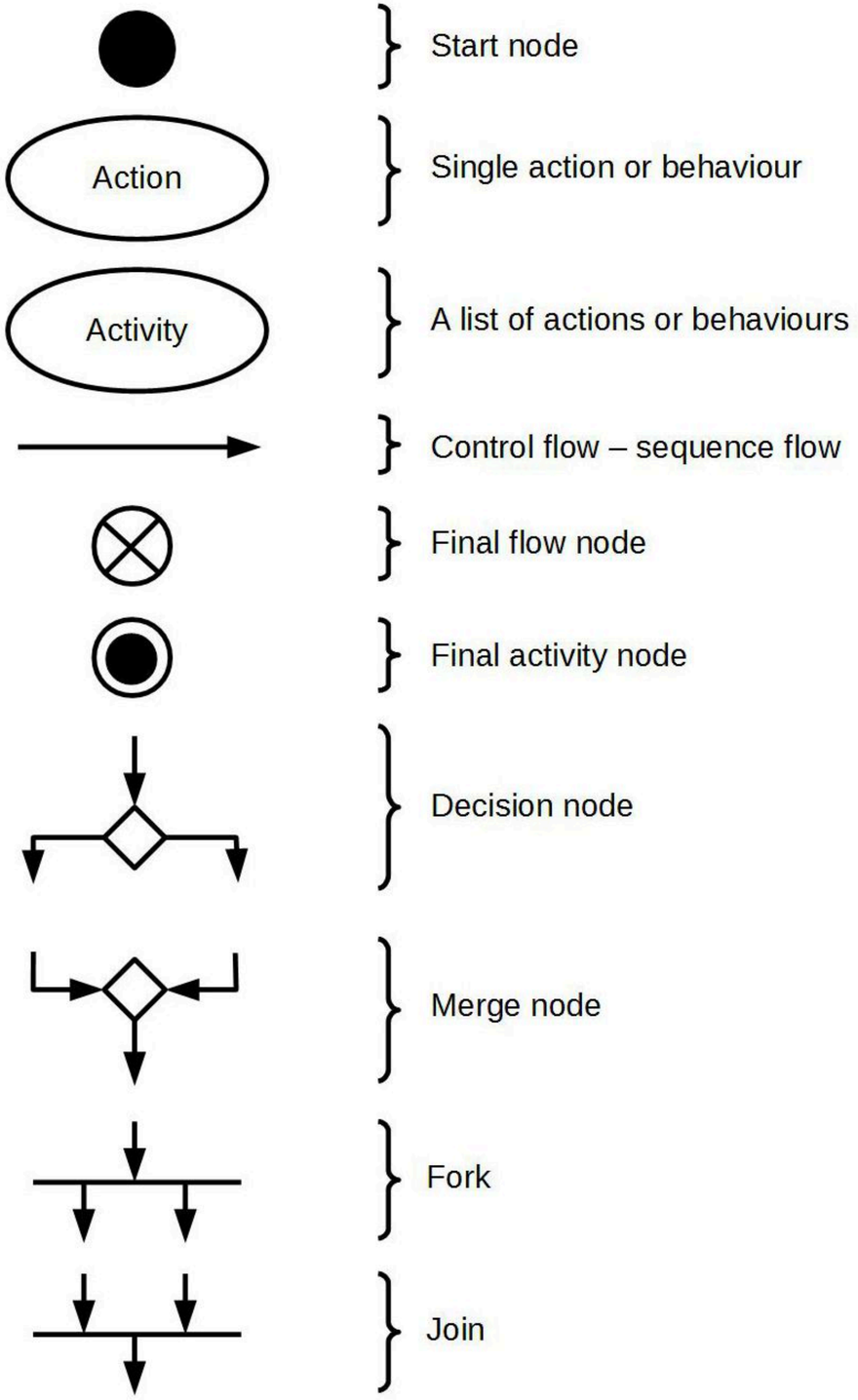


UML-diagram

Aktivitet/flödesschema

- Beskriver flödet i ett program inklusive beslutspunkter.

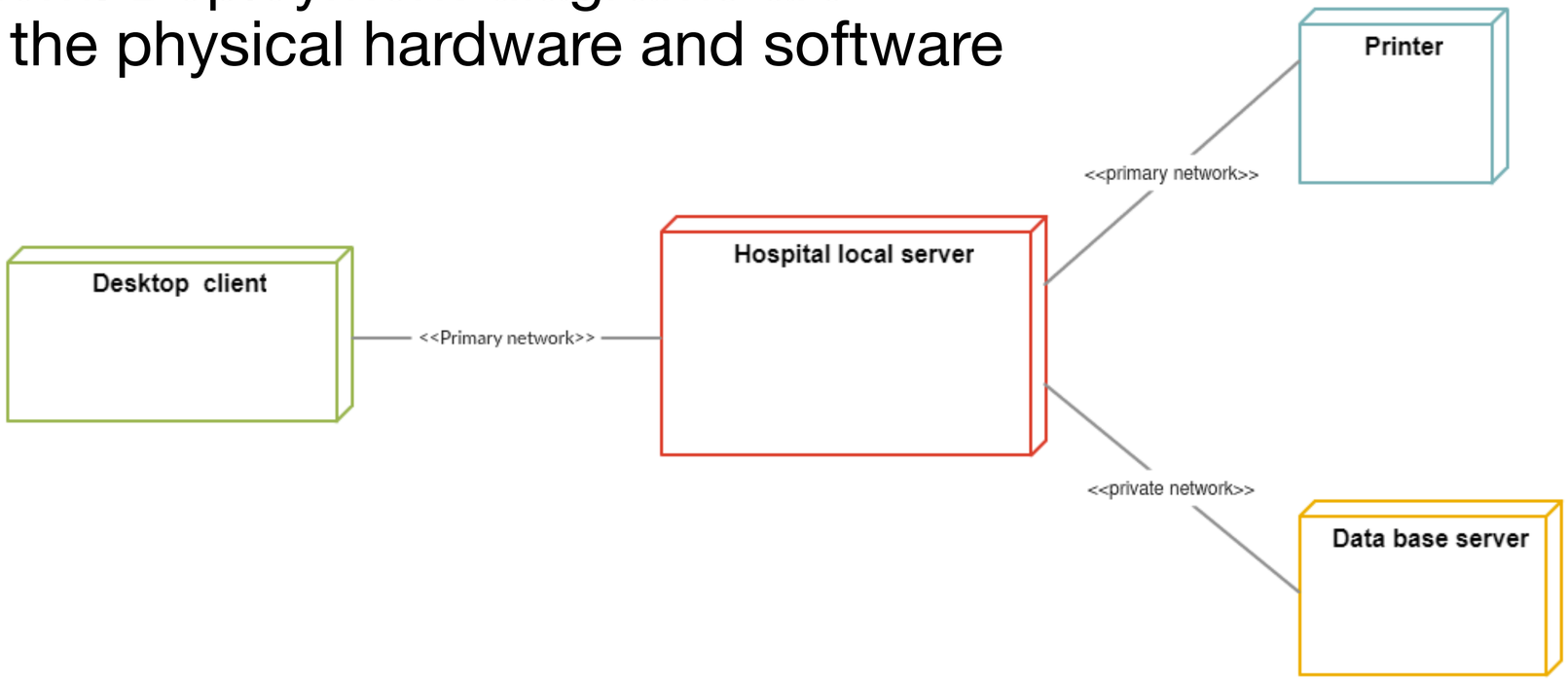




UML-diagram

Deployment diagram

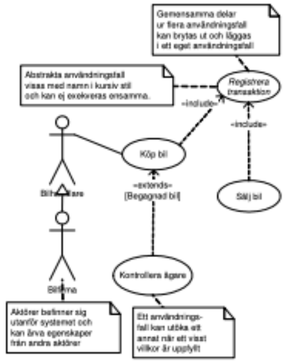
- A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.



UML-diagram
Lathund

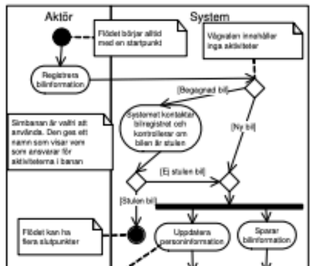
- http://www.acctest.systemvaruhuset.se/wp-content/uploads/2013/04/uml-lathund.pdf

1. Användningsfallsdiagram



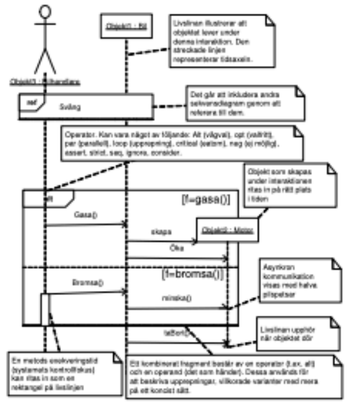
Ett användningsfall representerar en interaktion mellan systemet och en eller flera aktörer som resulterar i nytta för aktören. Användningsfallet används för att ge en övergripande beskrivning av systemets funktionalitet och vilken nytta kunden kan ha av det. Användningsfallet används för att fånga specifika, funktionella krav. Övriga krav dokumenteras i en kompletterande kravspecifikation.

2. Aktivitetsdiagram



Aktivitetsdiagrammet är ett traditionellt flödesschema. Det kan t.ex. användas för att specificera interaktionen mellan aktör och system i användningsfallet och/eller för att beskriva verksamhets-processer. Diagrammets fokus ligger på aktivitet-

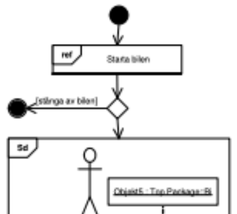
3. Sekvensdiagram



Sekvensdiagrammet används för att visa hur (system)objekt samverkar med varandra. T.ex. för att åstadkomma funktionaliteten i ett användningsfall Returmeddelanden är implicita, men kan också visas som en streckad returpil.

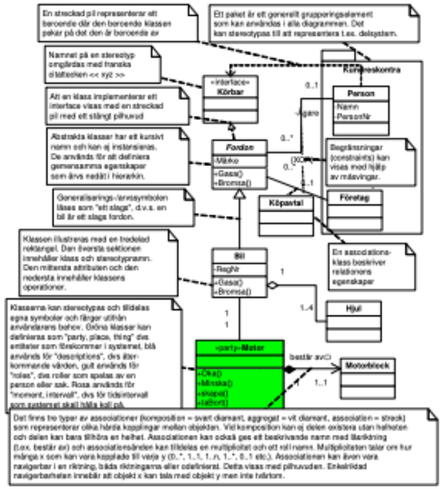
OBS OBS Sekvens-diagram ritas inte för alla interaktioner i systemet utan endast för de "intressanta"

4. Interaktionsöversiktsdiagram

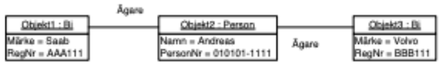


Interaktionsöversiktsdiagrammet är ett slags aktivitetsdiagram där man hänvisar till sekvensdiagram istället för att använda sig av aktivitets-noder. Antingen så ritas inter-aktionen in, inuti aktiviteten eller så hänvisar man till ett specifikt sekvensdiagram som beskriver interaktionen. Diagrammet är bra för att visa på samband mellan

5-6. Klass- och objektdiagram



Klassdiagramet är det vanligast förekommande och beskriver en struktur (system, data, information, mål, begrepp m.fl.). Det kan göras på olika nivåer (t.ex. implementationsoberoende-/analysnivå och språkspecifikt/designnivå). Objektdiagrammet exemplifierar klassdiagrammet med en ögonblicksbild.



Uppgift

- Ta fram exempel på use case-diagram, klassmodell, interaktionsdiagram och flödesschema för valfritt projekt, såsom inlämningsuppgifter, Yatzy-eller något av exemplen vi har jobbat med under kursen såsom Todo, Biblioteket eller annat.