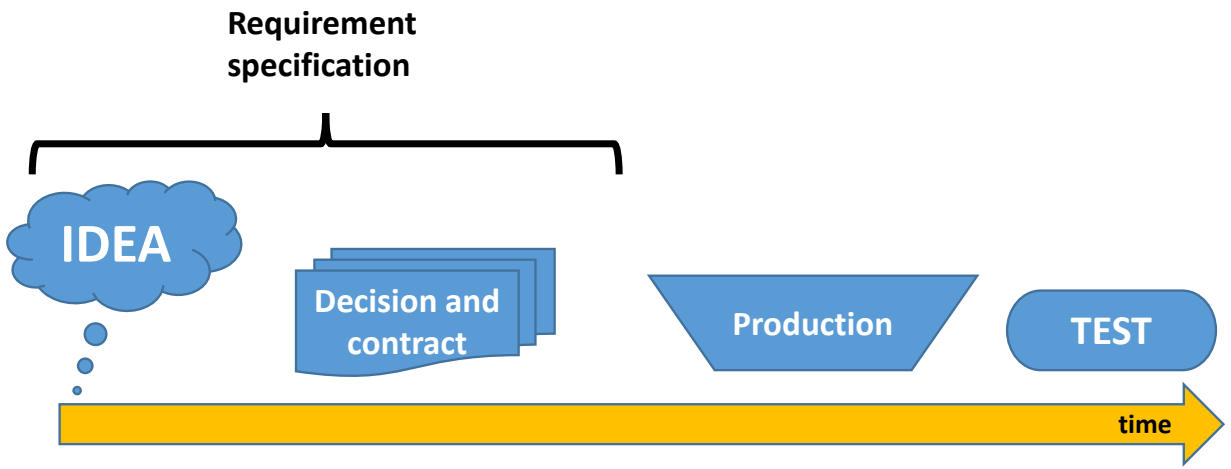
Torsdag 18-11-2021

PROCES-MODELLERING

Med

Aktivitetsdiagrammer (UML)

System development



What are we supposed to accomplich?

Understand customers intensions
Understand customers domain
Identify customers requirements
Formulate requirements

- Listen to the customer
- Help the customer to understand the possibilities

- **Document requirements**
- **Verify requirements**

Be ready to start developing the system

Tools:
Vision & Goals
Domain model, Activity diagram
Use Case diagram
User Stories
Mock Ups

Aktivitets diagrammer

- Aktivitets diagrammmer kan bruges til at:
  - Dokumentere eksisterende processer
  - Analysere nye processer
  - Dermed finde "reengineerings-muligheder"

AS IS

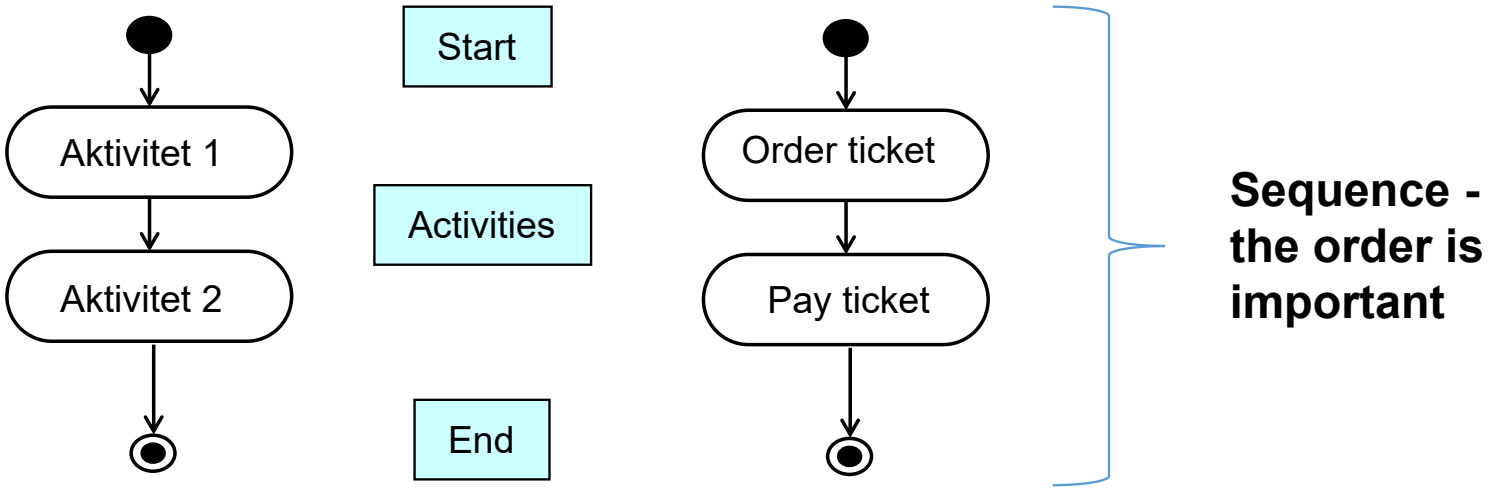
TO BE

- Diagrammet giver mulighed for at vise
  - Aktiviteter der indgår i en process
    - Sekvens
    - Parallel
  - Forgreninger
  - Valg (branch)
  - Artefakter
  - Ansvarsområder

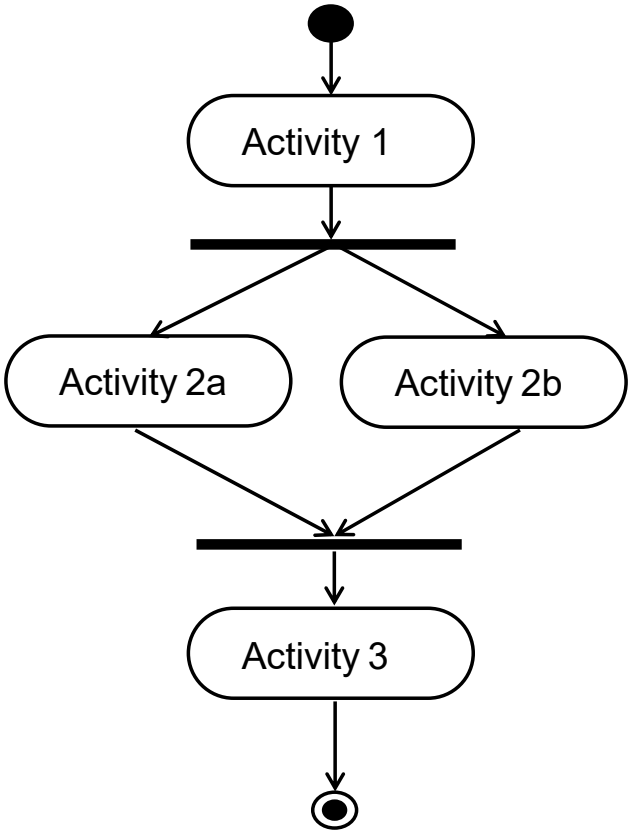
HUSK ALTID NÅR MAN MODELLERER:

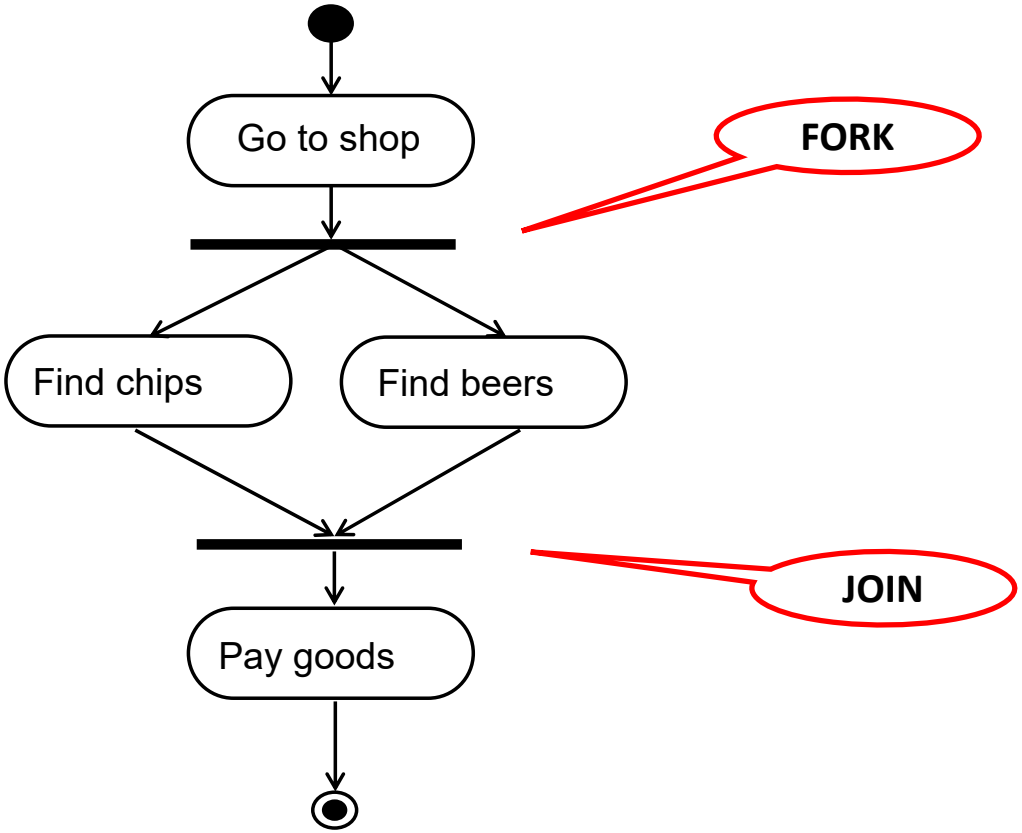
- **Fastlæg detaljeringsgrad**
  - **Formål/anvendelse**
  - **Tilgængelig info**
  - **Mulighed for "sub-diagrams"**

Aktivitets diagrammer - sekvens

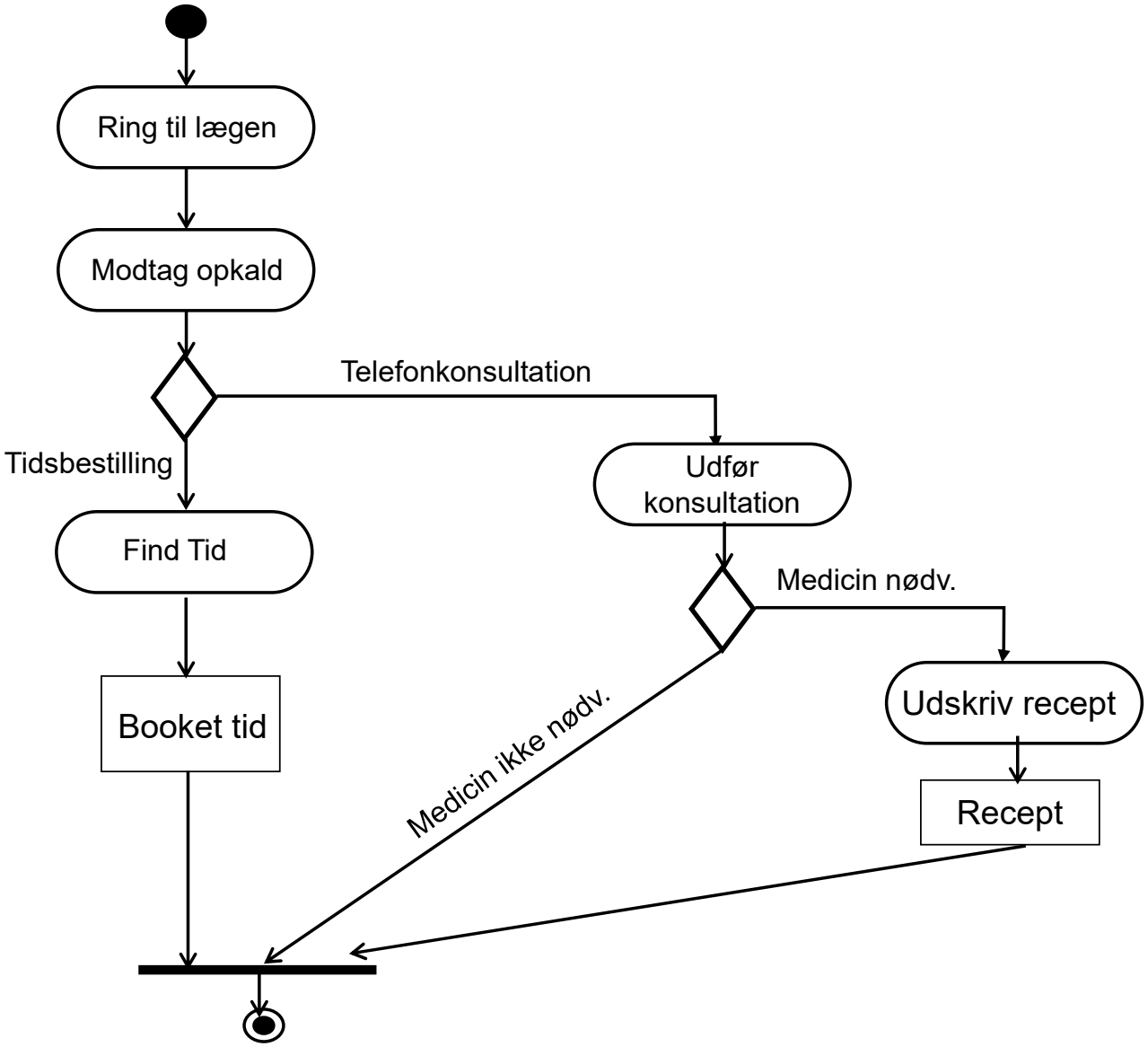


Aktivitets diagrammer – parallelle aktiviteter

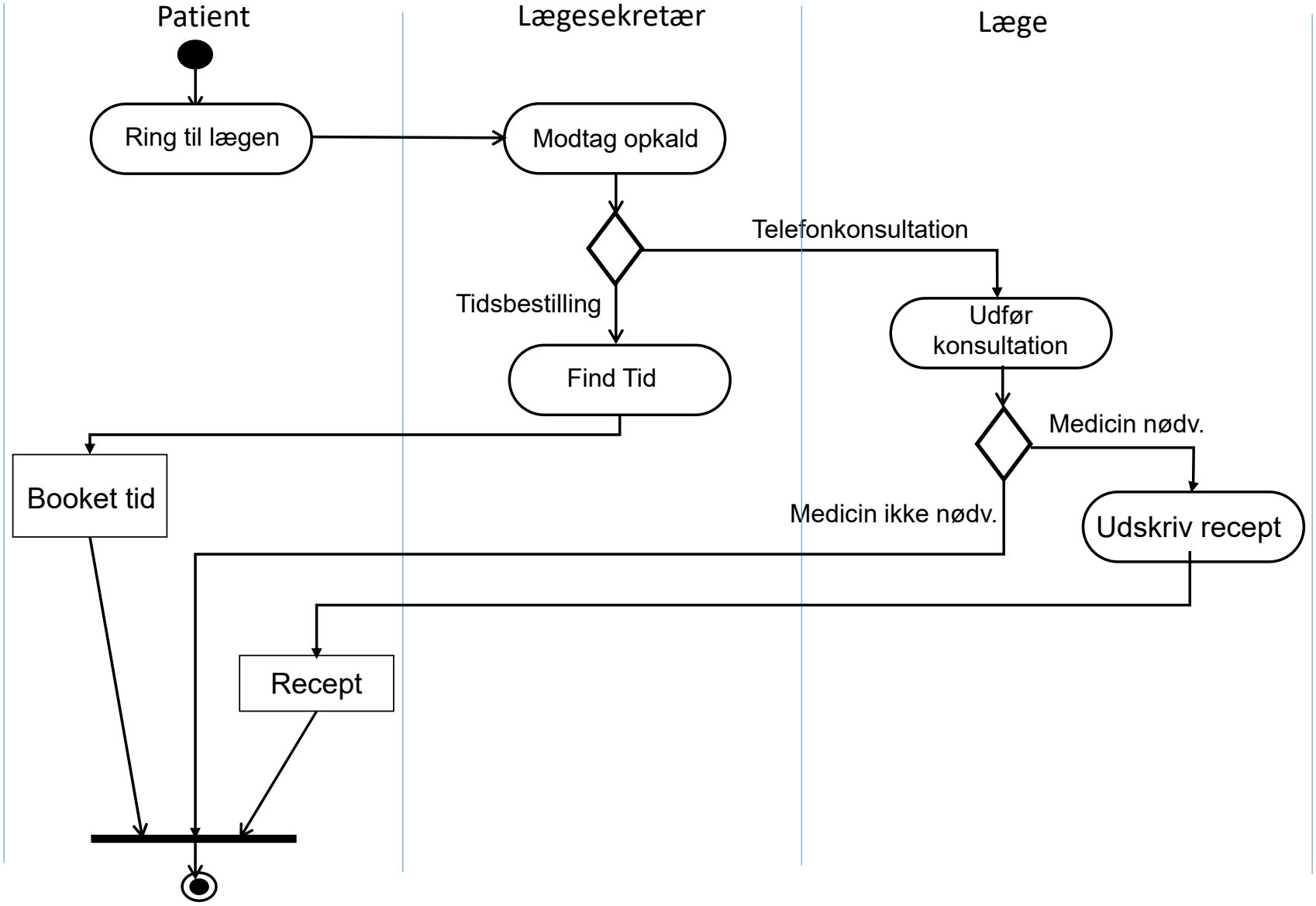




Aktivitets diagrammer – Valg (Branch)



Aktivitets diagrammer – Swimlanes



Aktivitets diagrammer – VIGTIGT HUSK

Altid markere START og SLUT

Alle veje gennem diagrammet skal gå til slut

Navne på aktiviteter indeholder et verbum i bydemåde

Artefakter vises i den Swimlane hvor de modtages og IKKE hvor de produceres

Øvelse (AS IS)

Lav et **aktivitetsdiagram med "swimlanes"** **(**HUSK: IT systemet skal have egen "swimlane"**)**

Quality assurance of software

In the company a task-owner can define tasks he want other to solve.

When such a task is defined, the IT-system automatically checks if the description contains information in all fields. If this is not the case, the IT-systems sends the task back to the task-owner who must revise the task.

If the task has relevant information the quality assurance person checks if the task is understandable. If this is not the case, the task is sent back to the task-owner. If the tasks description is understandable the quality assurance person adds a test to the task and sends task and test to a task-solver.

The task-solver estimates the time needed for handling the task and solves the task. After solving the task, the task-solver adds information about the used time.

The IT system moves the task and the solution to a tester who tests the solution. If the test fails, the task returns to the task-solver with information about test problems. If the test does not fail, the solution and the test are returned to the task-owner.

Øvelse (TO BE)

Lav et **aktivitetsdiagram med "swimlanes"** **(**HUSK: IT systemet skal have egen "swimlane"**)**

Quality assurance of software

In the company a task-owner can define tasks he wants other to solve.

When such a task is defined, the IT-system automatically checks if all the fields in the description are filled in. If this is not the case the IT-system shows a warning of missing fields to be filled with information.

If the task has all the necessary fields, the quality assurance person checks whether the task is understandable or not. If this is not the case, the task is marked "needs better info" and is sent back to the task owner through the IT-system. If the task description is understandable, the quality assurance person adds a test to the task and marks it "ready for solving" and the IT system sends it to a task solver.

The task solver estimates and notes the time needed for handling the task and solves it. After solving the task, the task solver marks the tasks as solved and adds information about used time.

The IT system moves the task and the solution to a tester who tests the solution. If the test fails, the task returns to the task solver marked as failed in test with information about test problems. If the test does not fail the solution and the test are returned to the task owner marked as solved and passed test.