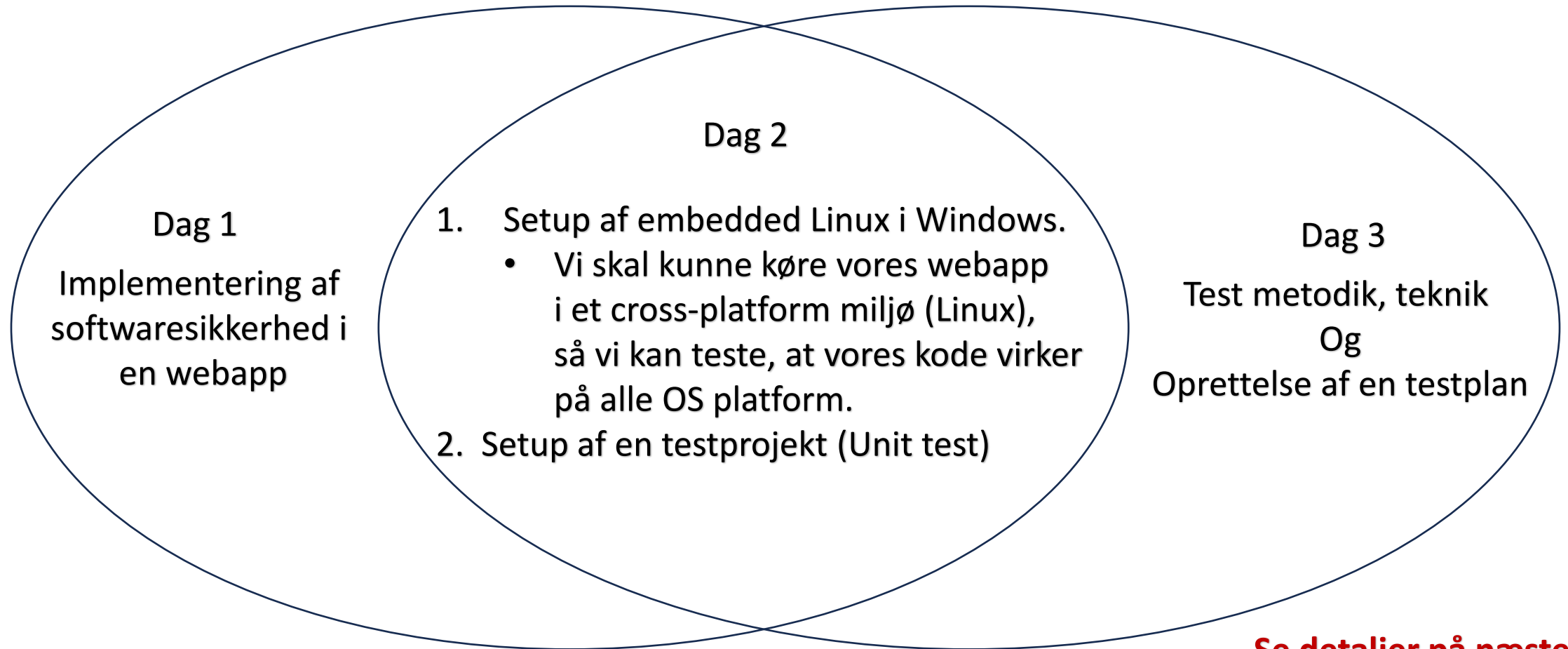


Setup af testmiljø

Testprojekt og kørsel miljø

Dag 2 program

Testmiljø



Se detaljer på næste slide

Øvelse 1 af 2

Setup af cross-platform testmiljø

Vi har prøvet “2 factor authentication” hvor vi selv skal generere token (2f kode) til brugeren (via e-mail eller SMS). Vi skal nu prøve **2f authentication med en authenticator**, hvor token auto genereres af authenticatoren.

- Vi skal download og installerer **Google Authenticator** fra Google Play på vores mobil device.
- Find og implementer i en passende lokation i vores webapp til **generering af en QR-kode image** som brugeren skal skannes til 2f registrering på authenticatoren. Image generering er typisk et punkt som adskiller de forskellige OS platform, derfor oplagt funktionalitet til en kompatibilitet test. Til dette skal vi

1. Aktiver WSL på PCen

- Kræver en IKKE HOME EDITION Windows PC (lån gerne skolens PC)
 - Udvikler man på Mac eller Linux maskiner I forvejen, er det omvendt, hvor man tester istedet mod Windows. Og det gøres normal med en Virtual Maskine med Windows OS.

2. Installeringer

- Installer Linux (Ubuntu, command-line utility version)
 - I må gerne installer det “stand-alone”, og lade Visual Studio opdage den. Men jeg gennemgår installation via Visual Studio.
- I Linux, Installer .NET

3. Identificering af componenter som kræver **Mocking** for at kunne køres under Linux (At isolerer kode/component under test: **CUT**. **CUT** i unit test → **code under test**, her → **component under test**).

- Vi har kun en komponent som er ekstern i vores web app lige nu: **Databasen**. Kræver den mocking er afhængigt af hvilket database der er valgt!

Øvelse 2 af 2

Setup af automation testmiljø (Unit test)

Code-along øvelse:

- Testprojekt type som skal oprettes er: **xUnit**. Ude fra det web app vi har implementeret, har vi følgende komponenter som vi kan test på:
 - **Test komponent view**
 - **Test af kode i komponentens 'code' sektion**
 - **Test af authentication (authenticated/not authenticated)**
 - **Test af authorization. Vi har implementeret en admin rolle, så vi skal test (er admin/ er ikke admin)**
 - **Test af C# klasser**

Der kan være mere/andet at teste, afhængigt af hvordan i har fortolkede 'casen' til aflevering og hvad i har valgt at implementeret. Gennemgang af punkterne foroven skulle gerne ruste jer til selv at kunne finde og fortolke eksemplerne med Google som resource.