**Opgave 1**

1. **Hvad er en produkt-backlog i forhold til Scrum og hvordan bruges den?**

I Scrum er produktbackloggen alle ens requirements, samt en liste af alt det arbejde man ønsker at få ordnet til projektet. Hver element på listen har en prioritetsværdi (så man kan se hvilket arbejde der er vigtigst). Disse prioriteter bliver sat af produktejeren og de bliver reprioriteret ved starten af hvert sprint.

1. **Hvad er forskellen på en ”black-box” og ”white-box”**

I en white-box test får man lov til at se kildekoden, og dermed også hvordan programmet er lavet, hvilket betyder at man kan lave gode (dog uinteressante) tests. Ved black-box testing har man ikke adgang til kildekoden, og må derfor nøjes med at vælge gode input og se hvad outputtet resulterer i, hvilket afspejler hvordan en slutbruger ville opleve programmet.

1. **Beskriv kort formålet med at udarbejde en domænemodel og hvad den beskriver.**

Man bruger en domænemodel til at skabe et strukturelt overblik over et problems/opgaves domæne. Man identificerer de konceptuelle klasser, deres relationer mellem hinanden samt deres attributter.

1. **Forklar kort hvilken betydning begreberne ”coupling” og ”cohesion” har i forhold til et godt arkitekturdesign.**

Kobling betyder hvor mange andre elementer et systemelement direkte arbejder sammen med (f.eks. med metodekald osv.). Hvis der er alt for høj kobling i programmet, bliver det uoverskueligt (stereotypisk spagettikode), hvilket gør det langt sværere at debugge, vedligeholde og modificere. Cohesion eller på dansk samhørighed bruges til at betegne hvor godt rollerne er fordelt blandt softwareklasserne. F.eks. skal en klasse *netværk* i telefonsoftware ikke håndtere medieafspilning eller datastrukturer, men derimod håndtere netværket. På denne måde ender man med et program hvor hver enkelt metode sørger for en given funktion, som igen gør koden letlæseligt og nemmere at forbedre.

1. En applikationsmodel kan illustreres vha. et UML/SysML Klassediagram. De består af hhv. domæneklasser, grænsefladeklasser og kontrolklasser. Kontrolklasserne repræsenterer funktionaliteten beskrevet i use case diagrammet, og er derfor navngivet det samme som use casen. Grænsefladeklasser er alle de klasser hvor der kan komme input udefra (som f.eks. en brugergrænseflade eller en kortlæser) – dvs. aktørerne. Domæneklasserne er resten, hvilket er domænerne.