

Agile HW/SW co-design of safety-  
critical solutions:  
new research opportunities

May 30, 2016

# Objectives

- Keynote and discussion on agile methods applied to HW dev. – in sync. with SW
- A short introduction to the SafeScrum initiative
- Outline ideas and opportunities for a new R&D project, addressing key challenges in the domain of safety-critical systems (SCS) development
  - Security challenges in SCS
  - Agile HW/SW co-design
  - Life-cycle management of SCS
  - Formal methods

# Program

<b>Part 1</b> Agile HW/SW co-design of safety-critical systems	Simple breakfast and mingle	All	0830-0900
	Welcome – objective – agenda	SINTEF	0900-0915
	Keynote: Yes, hardware can be agile	Nancy Van Schooenderwoert (Lean-Agile Partners)	0915-1015
	Coffe break and mingle	All	1015-1030
	Product development using SafeScrum	T. Stålhane	1030-1100
	Kompetanseprosjekt for næringslivet	G.K. Hanssen	1100-1200
<b>Lunch break</b>			1200-1300
<b>Part 2</b> Joint R&D	Co-design Safety&Security Livs-syklus drift & formelle metoder Formelle metoder	D. Rognlien, G.K.Hanssen B. Haugset, S. Petersen, S.O. Johnsen T. Myklebust Ø. Teig, F. Aakvik	1300-1400
	Preliminary planning	All	1400-1500

# Co-design – noen utfordringer

(Co-design; koordinering av HW og SW-utvikling)

- Konstruere og verifisere systemet før man har landet HW/SW partisjoneringen
- Utsette tidspunkt for partisjoneringen så lenge som mulig, og å velge grense for partisjonering basert på systemet
- Håndtere programvarekonstruksjon og verifikasjon før HW er tilgjengelig
- Være trygg på at HW og SW konstrueres riktig i forhold til hverandre når dette gjøres samtidig
- Oppnå effekt av smidige prinsipper
  - Frequent delivery of the most valuable working software and hardware (design)
  - Close collaboration and frequent communication, (btw. HW/SW teams, and problem owner)
  - Self-organizing teams that learn as they collaborate
  - Handle changing requirements without crisis.
- Smidig *system*utvikling iht. viktige standarder

# Co-design - idéer

- Smidig utvikling av embedded-systemer, muligens også FPGA-kode
- En del kode må utvikles, ikke for å implementere funksjonalitet, men for å sette de rette omgivelsene.
- Mye kode må skrives for å konfigurere MCU riktig, sette opp rett konfigurasjon av omliggende enheter, håndtere generelle ting som timing, minneallokering, adressemapping, DMA systemer. Særlig på en ny platform er dette en betydelig mengde kode.
- Tidligfase utvikling (før HW)
  - HW mock-ups for tidlig testing av SW, f.eks. driver stubs eller emulering driver-driver
  - Bus Functional Models
- Hvor mye må designes tidlig?