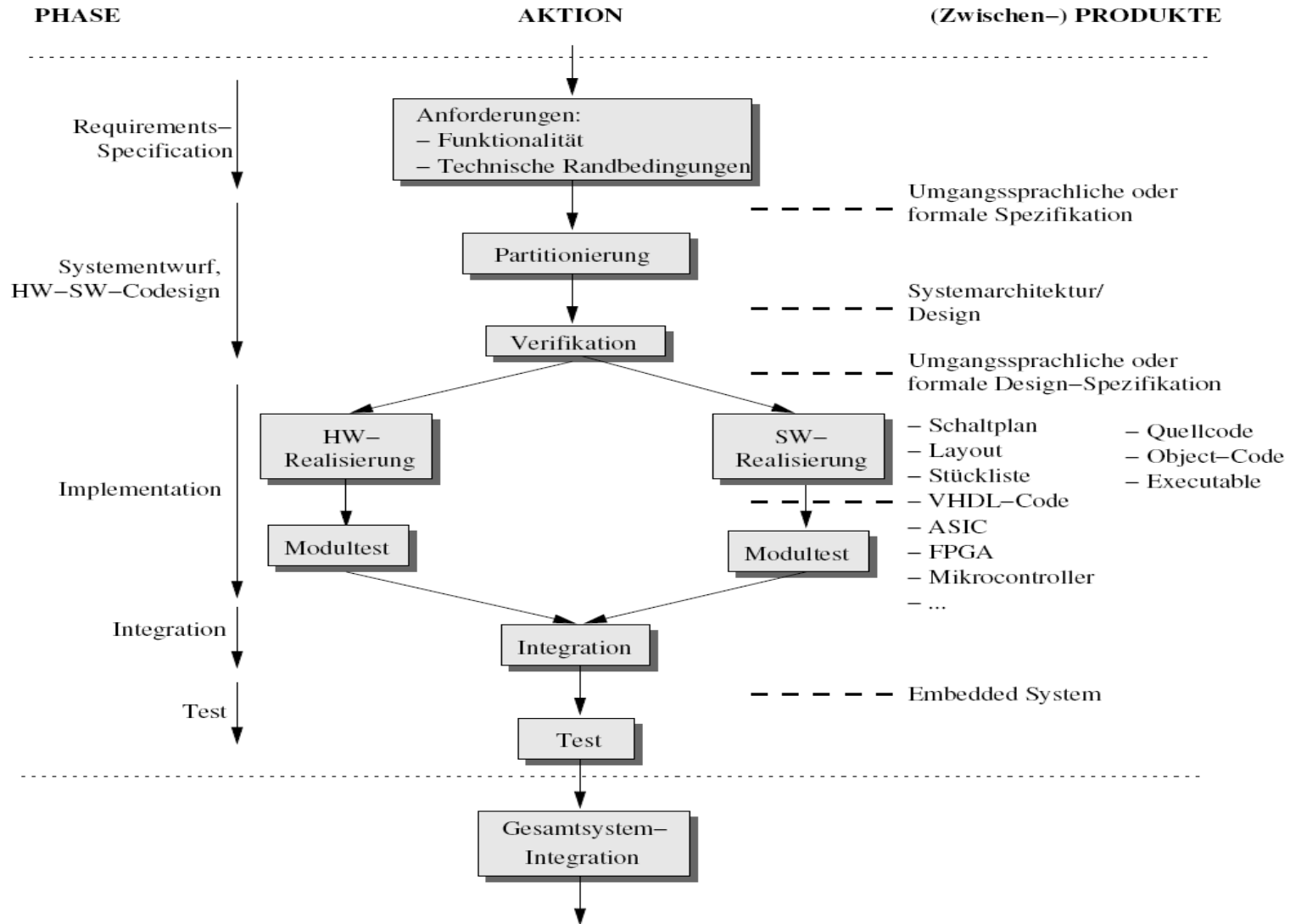
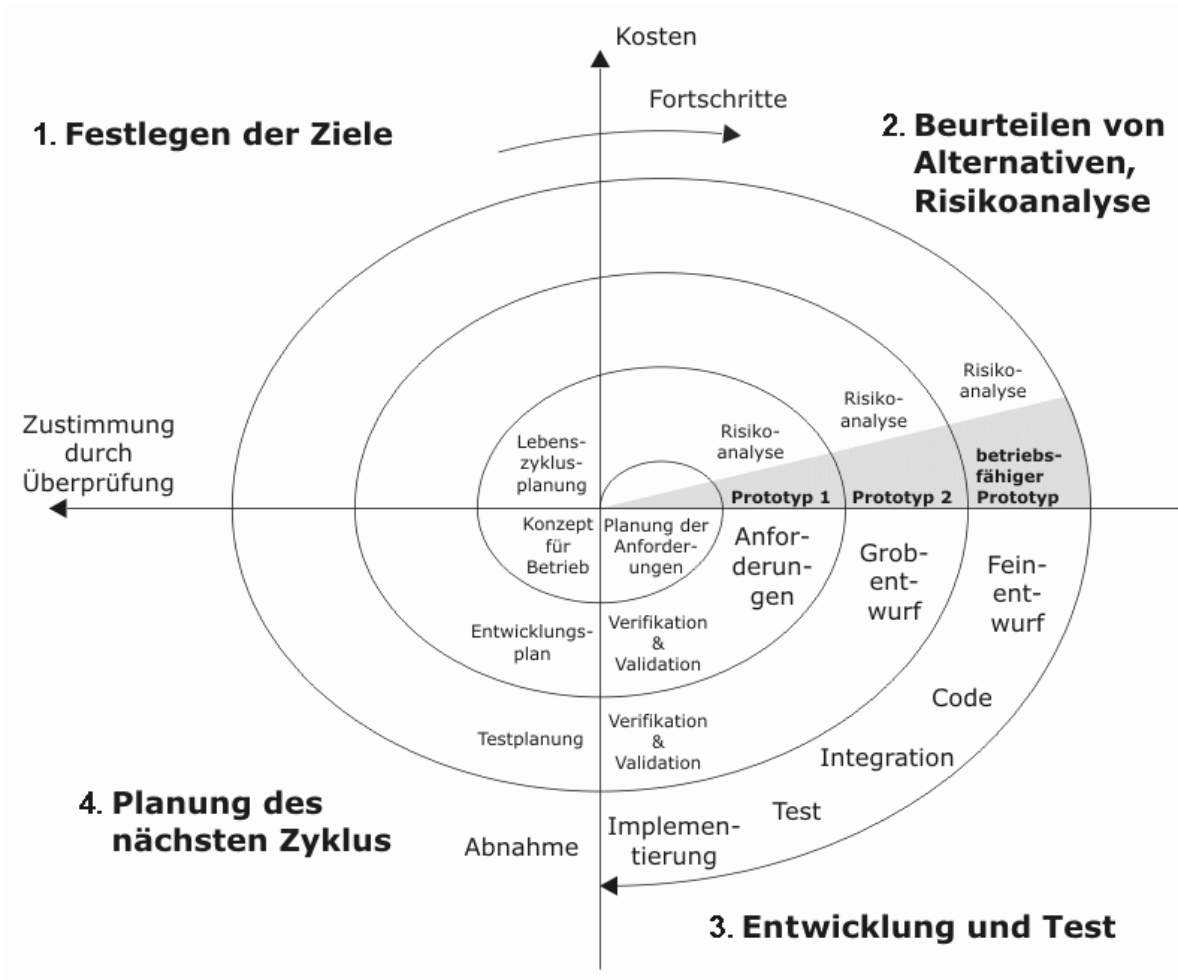


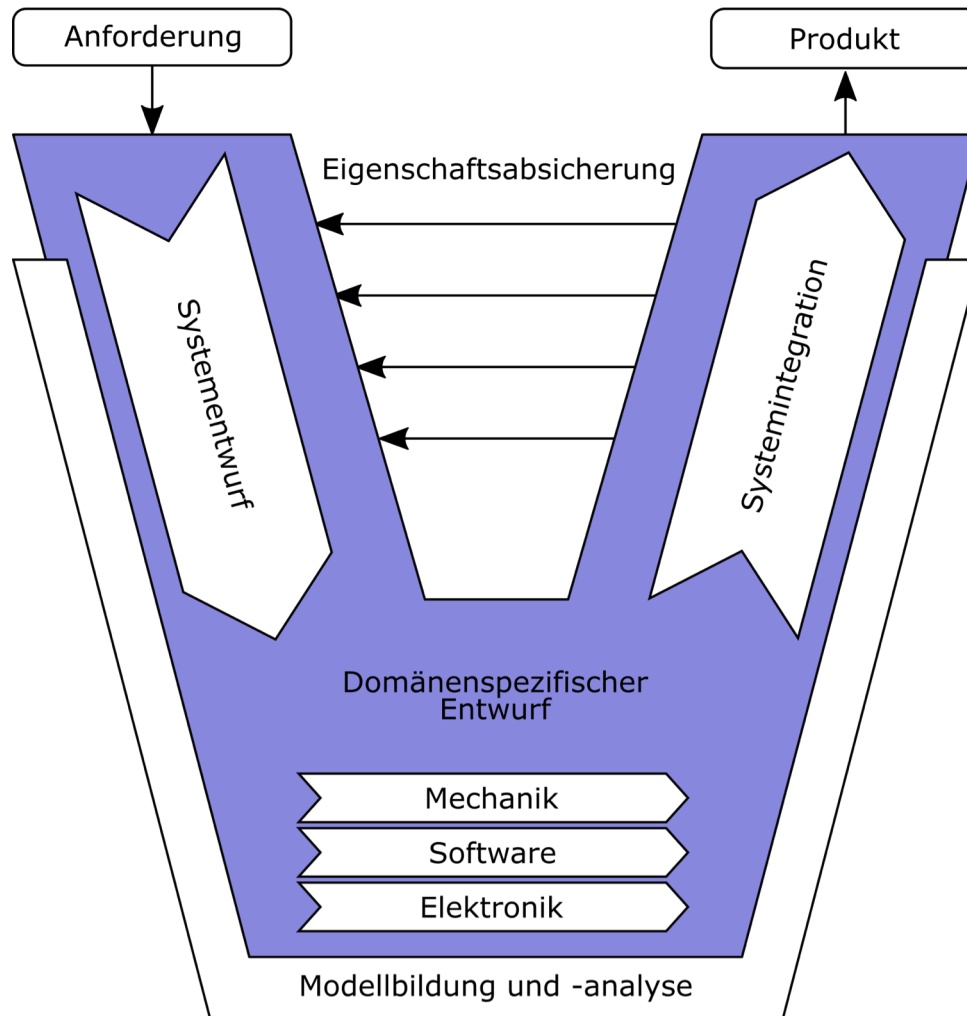
# Systementwurf (klassisch)



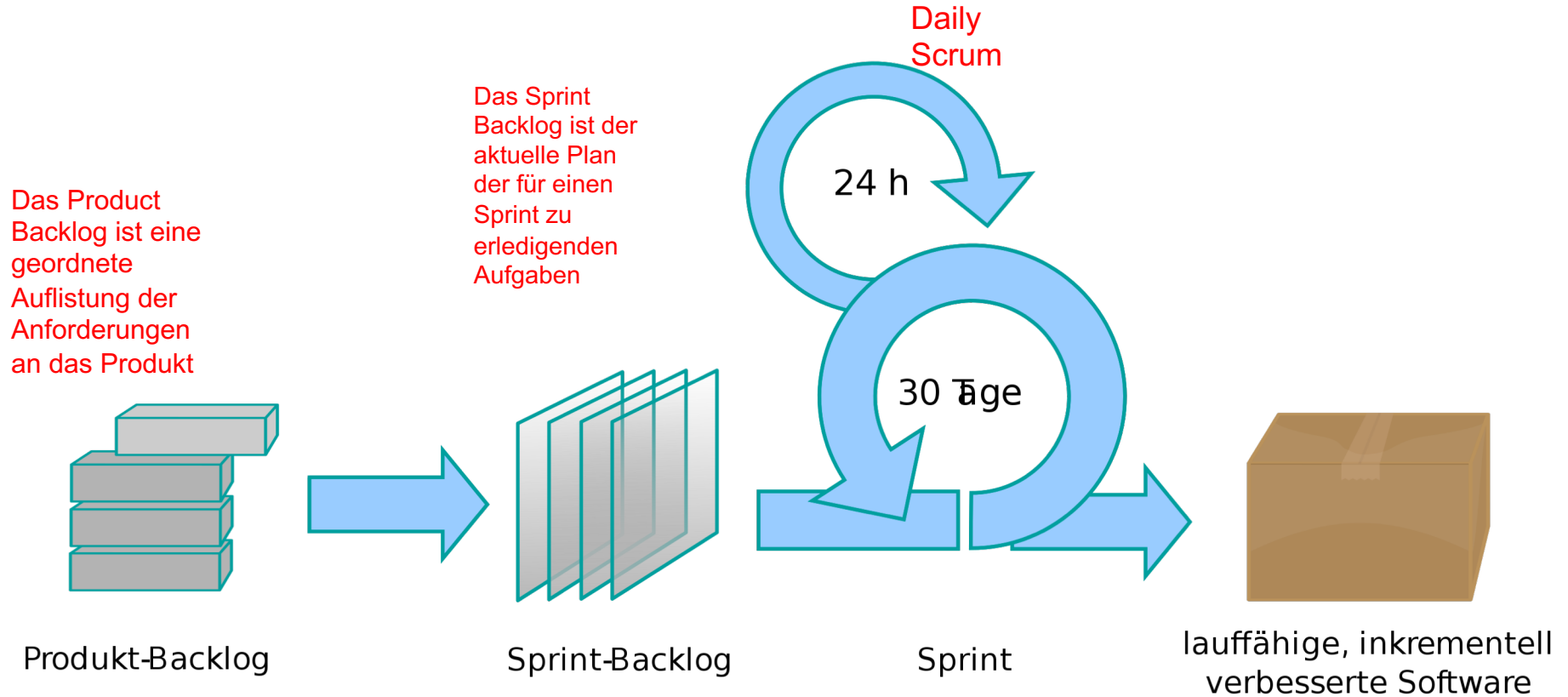
# Systementwurf (z.B. Spiralmodell)



# Systementwurf (V-Modell XT)



# Systementwurf (?Scrum)



# Systementwurf

## Systementwurf, ganzheitliche Betrachtung

Falls kein Einsatz von Standard-Komponenten (SPS,...) möglich:

**ganzheitlicher Entwurfsprozess zur Erfüllung sämtlicher Anforderungen**

- **Entwurfsprozess umfasst sowohl die Entwicklung der Hardware als auch der Software**
- **Entscheidung während der Entwurfsphase, welche Funktionalität durch Hardware und welche durch Software realisiert wird (*Hardware/Software Codesign*)**
- **Möglichkeiten, Funktionalität zwischen Hardware und Software zu verschieben**



# Systementwurf

## Host-Target Entwicklung

### „Self-Hosted“ Entwicklung

Entwicklungsumgebung und Zielsystem sind identisch (so wie wir alle es kennen)

Manche RT-BS (LynxOS, QNX) sind mächtig genug, um Entwicklungssystem zu unterstützen

Aber: Hardware einer Prozesssteuerung meist

- proprietär
- zu leistungsschwach (Rechenzeit, Speicher, Peripherie)
- keine geeignete Systemsoftware vorhanden

Folge: Entwicklungssystem „passt“ nicht auf das Zielsystem

Eigenes Entwicklungssystem

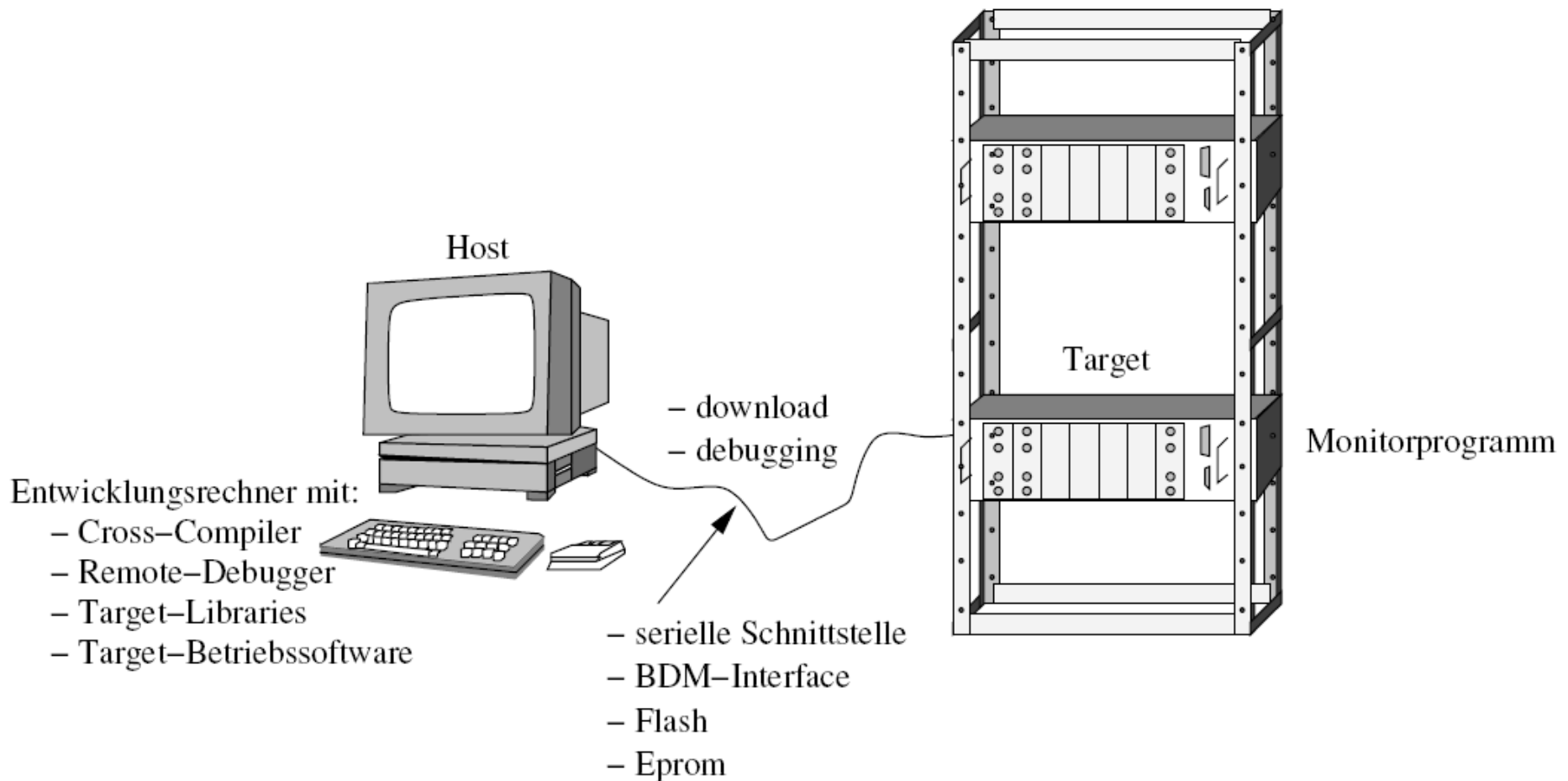
Entwicklungsrechner wird als *Host* bezeichnet

System, für das entwickelt wird, als *Zielsystem* oder *Target* (Host/Target Entwicklung).



# Systementwurf

## Host-Target Entwicklung



# Systementwurf

## Host-Target Entwicklung

Das wesentliche Element der Entwicklungsumgebung auf dem Host: **Cross-Compiler**

- **Läuft auf einer Plattform (z.B. Linux, X86), erzeugt Code für andere Plattform (z.B. QNX, Motorola PPC).**
- **Erzeugt „Image“:**
  - **Eigentliche Applikation**
  - **Betriebs- oder Laufzeitsystem - und**
  - **als Teil von diesem: Bootcode (Startupcode)**





# Systementwurf

## Host-Target Entwicklung

Während der Entwicklung: “Entwicklungstarget”

Ermöglicht das einfache Laden der neuen Software

Im Flash-EEPROM des Target: (Boot-) Monitor,

- einfaches Systemprogramm, über das Software geladen und gestartet werden kann.
- Oftmals bietet das Monitorprogramm auch einfache Debugmöglichkeiten.

Laden der Software: Code (Image)

- über Netz ( DHCP, TFTP, BOOTP)
- über eine serielle Schnittstelle
- in einen Baustein (z.B. EPROM) brennen (Lösung für das Produktivsystem),
- eine eventuell vorhandene Flashdisk programmieren
- vorhandene Schnittstellen (bei Motorola zum Beispiel das BDM-Interface, JTAG) zum in-System-Flashen



# Systementwurf

## Host-Target Entwicklung

### Target-Debugging mit Remote-Debugger:

- **Auf dem Host:**
  - GUI
  - Symboltabelle
  - Debug-Info (z.B. DWARF) aus dem Image
- **Auf dem Target: Server für**
  - Aktuellen Systemzustand (Stack, Variablenbelegung)
  - Breakpoints

### Verbunden über

- **Seriell, TCP/IP**
- **oder JTAG, BDM: spezielle Debug/Flash-Interfaces am Prozessor)**

### Ohne Remote-Debugger: Debugging mit

- **printf (über serielles Interface)**
- **DP-RAM**
- **IO-Leitungen, LEDs**



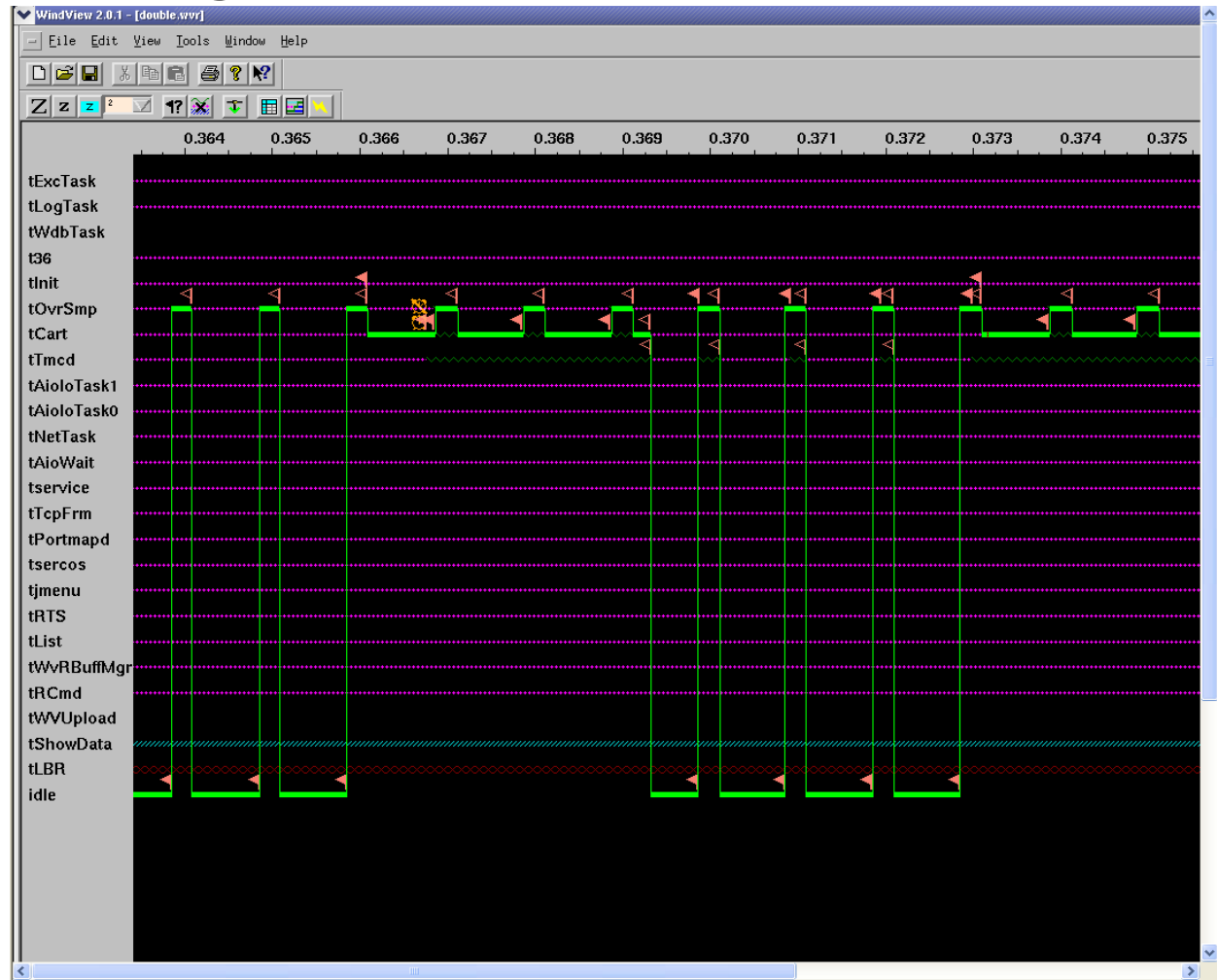
# Systementwurf

## Host-Target Entwicklung

Event-Browser/  
Kernel-Event-

Tracer:

- Signale,
- Task-Zustände,
- Semaphore,
- Ereignisse,
- Interrupts,...

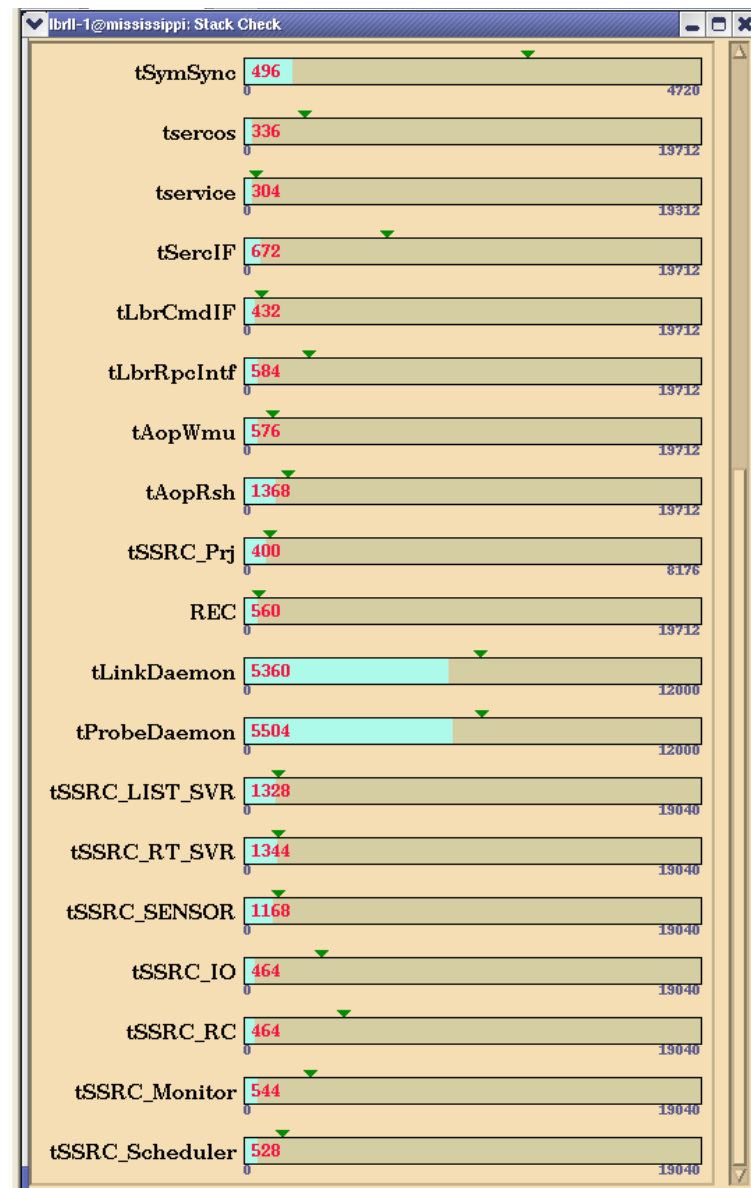


# Systementwurf

## Host-Target Entwicklung

### Stack-Monitor

- Maximal verfügbarer Stack/Task
- Aktuell
- „Hochwassermarke“

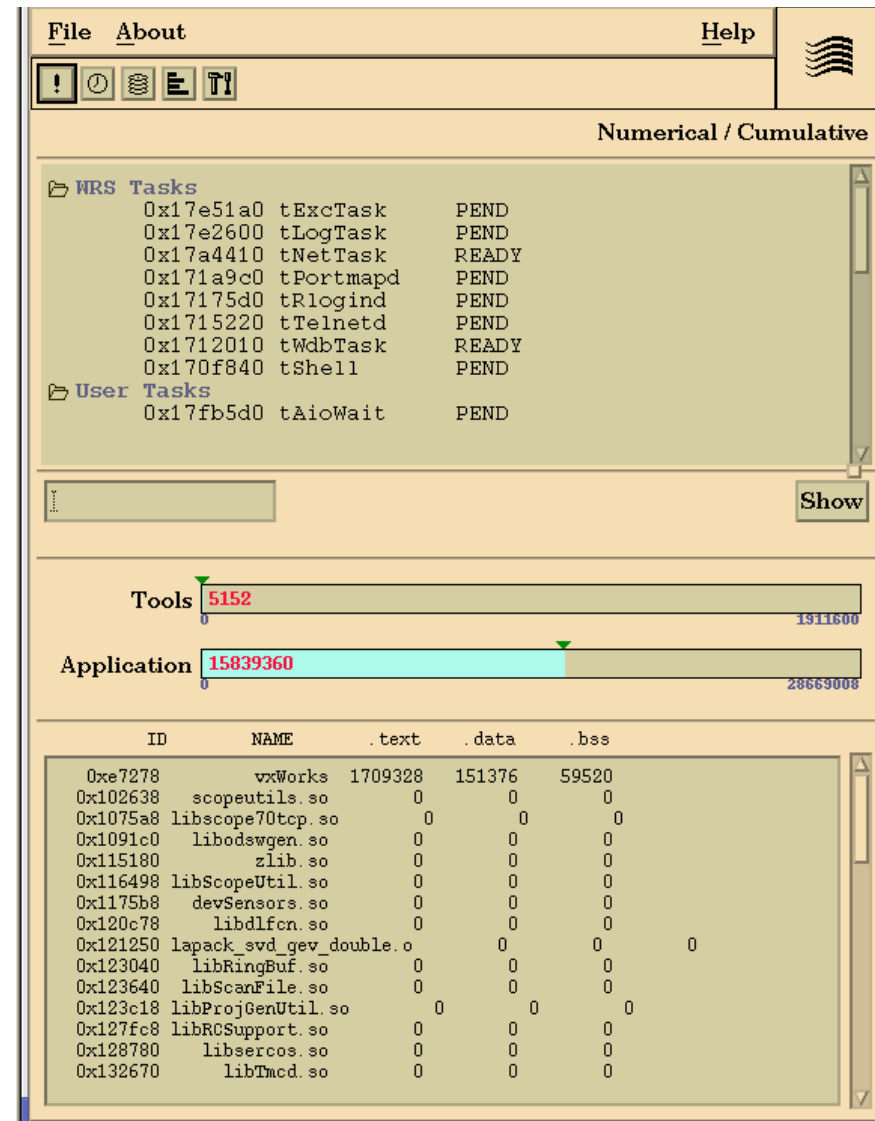
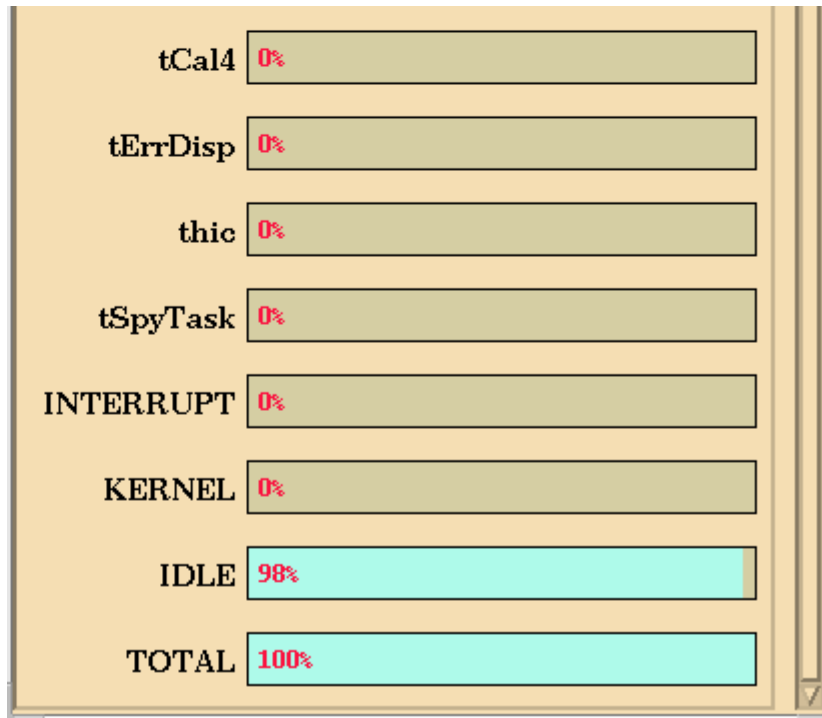


# Systementwurf

## Host-Target Entwicklung

### Speicherbelegung

## Auslastungsmonitor



# Systementwurf

Anforderung	Hardware	Software
Leistungsaufnahme	Bausteinauswahl	Sleepzustände, "Runtertakten"
Größe	Bausteinauswahl/Integrationsgrad	Verlagerung von HW in SW
Gewicht	Technologie/Bausteinauswahl	Verlagerung von HW in SW
Betriebsbedingungen	Bausteinauswahl (MIL-Varianten), Kühlung	(proprietäre Maßnahmen)
(Verkaufs-) Preis	Standard-Komponenten	Standard-OS, -Appl., -Protokolle
Betriebskosten	Technologie, Bausteinauswahl	(Sleepzustände, "Runtertakten")
Realzeitverhalten	Verlagerung von SW in HW, Rechenleistung erhöhen (z.B. DSP)	Realzeitsoftware, Algorithmen
Verfügbarkeit	Redundanz, Betriebsbedingungen, mechanische Maßnahmen	Redundanz
Sicherheit	Redundanz (2 aus 3 System)	Redundanz, Diversifikation
Langlebigkeit	Bausteinauswahl, Betriebsbedingungen	(Sleepzustände, "Runtertakten")
Wartungsaufwand	Modultechnik, Flash	Software-Upload

ise

