



## Development of SDL-Based Software for an Embedded System – Practical Experiences

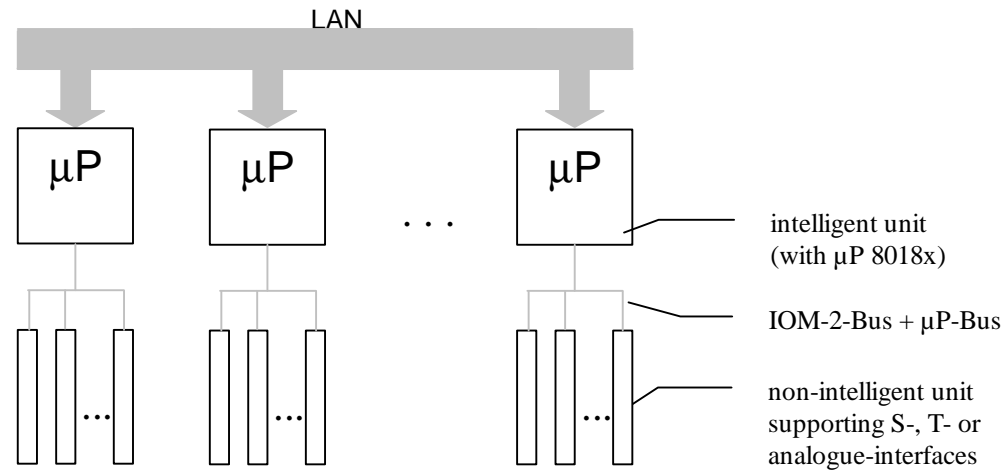
Authors: Stefan Bläsius, Josef Maier, Stefan Karg, Günther Kohler

Speaker: Günther Kohler

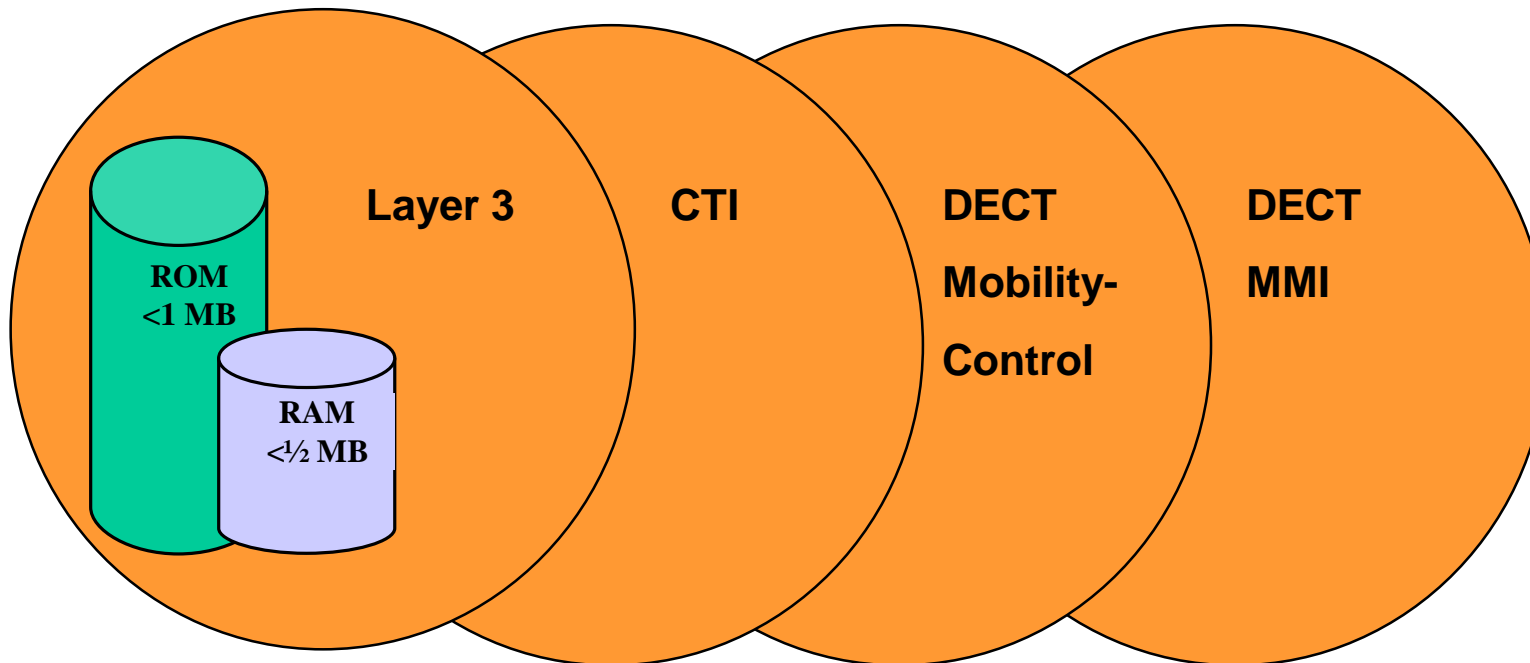
[www.tenovis.com](http://www.tenovis.com)

- **Target platform and its restrictions**
- **starting with SDL-88 (in-house developed tool)**
- **development process and quality ensurance using SDL & MSC**
- **host test and target feedback**
- **target testing with Telelogic's Microtester**
- **Development experiences using SDL**
- **Conclusion**

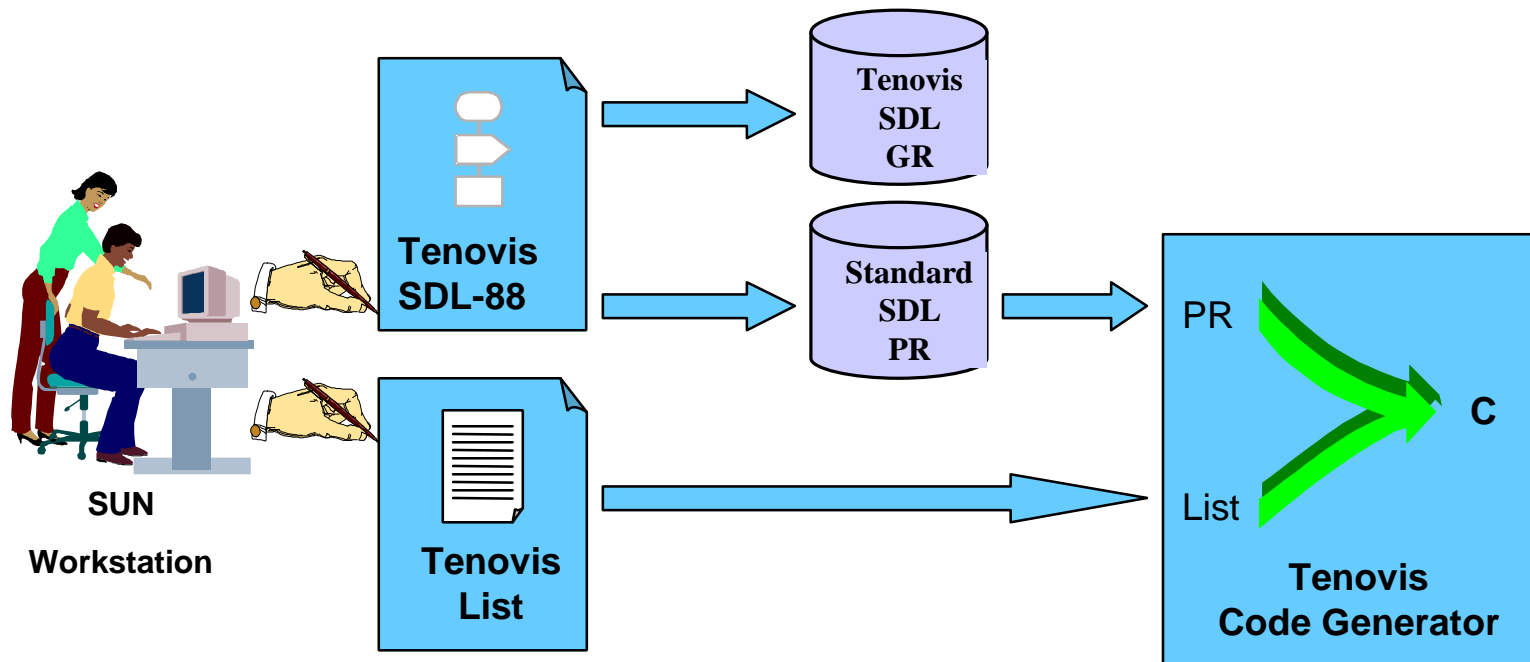
## System Integral 3



## SDL-Tasks and Target Restrictions



## In-house developed SDL-88 tool

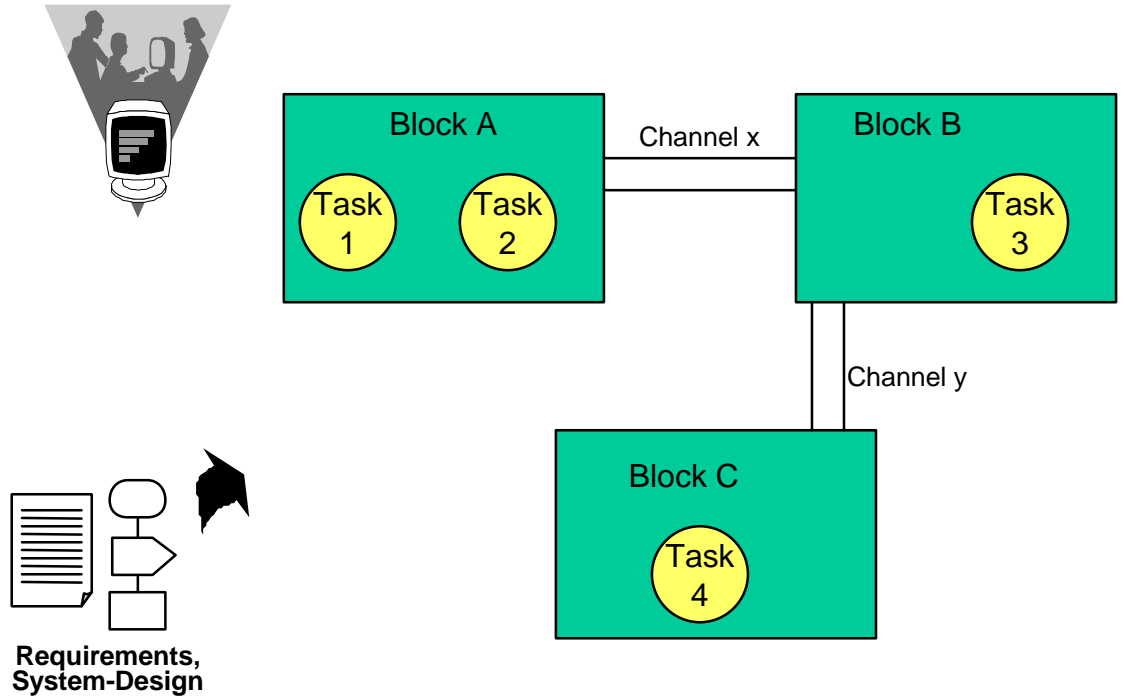


### **Drawbacks of our in-house SDL tool**

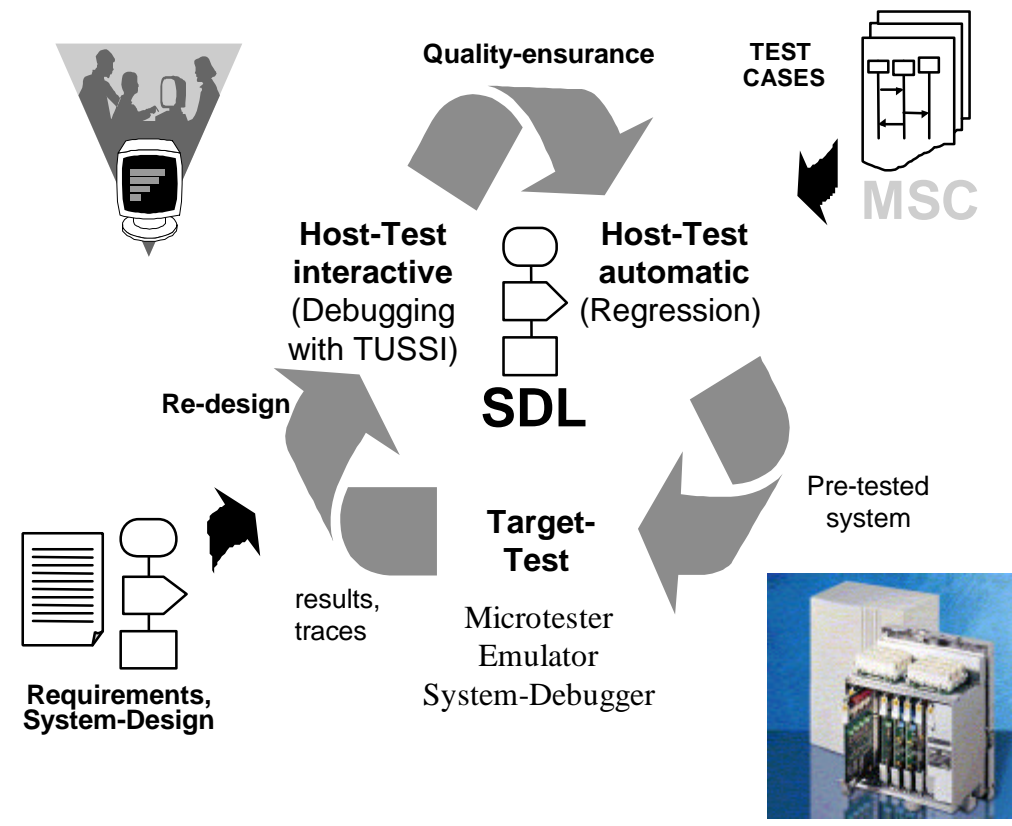
- **inter-process communication not supported**
- **process instantiation not supported**
- **insufficient tool chain**

**=> 1997: turn to Telelogic's SDT**

# development process and quality assurance using SDL & MSC



# development process and quality assurance using SDL & MSC



Development of SDL-Based Software - Practical Experiences

Presenation on SAM2000

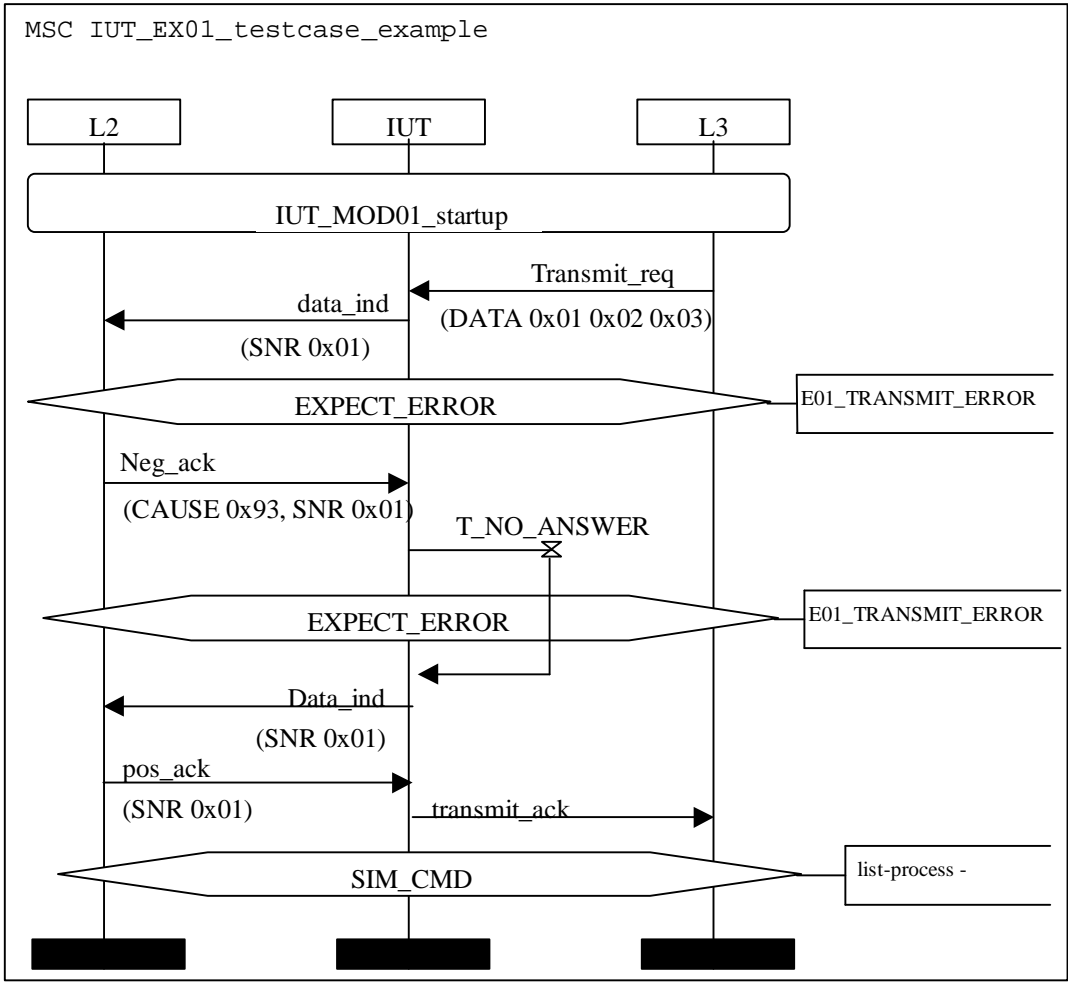


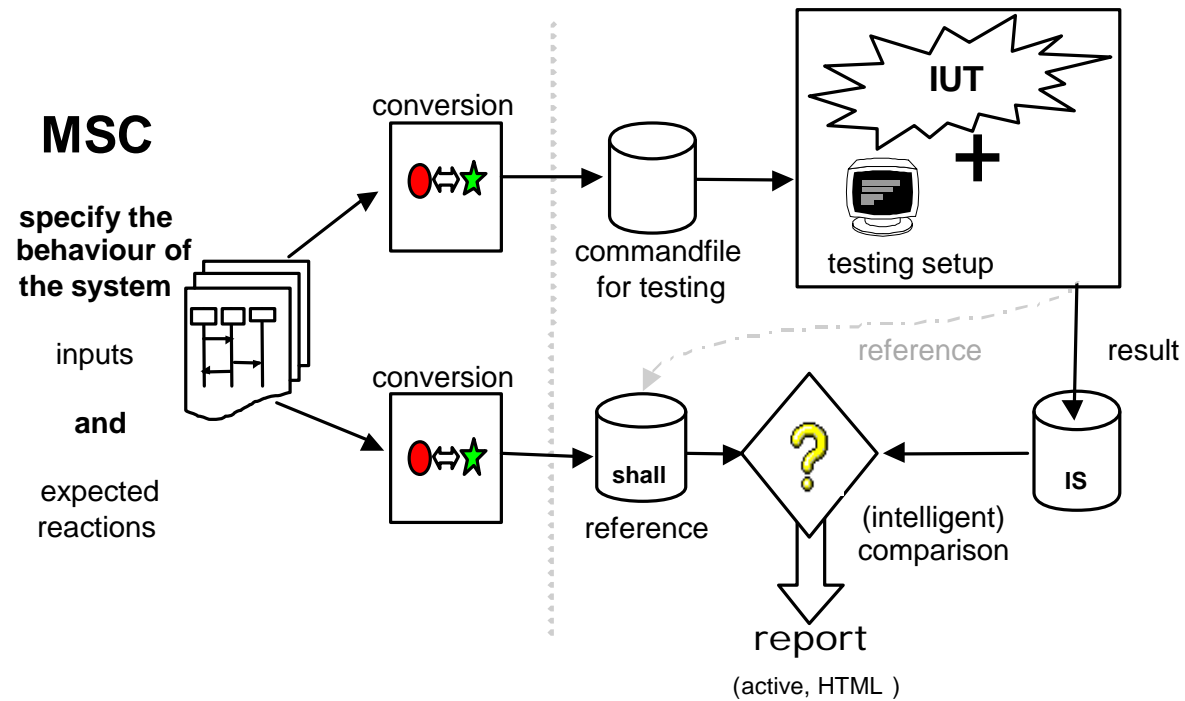
# Specifying Test-Cases by Using MSC

~HTML~  
 Test-case :  
 IUT\_EX01\_testcase\_example  
 Responsible : Stefan Karg /  
 Günther Kohler

In this example the syntax and semantics of the test-case designed by MSC are demonstrated.

~HTML~





# Test Report

Netscape: Testergebnisse für Projekt cti

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: file:///vobs/i3/sdt/cti/test/sun\_cu/reports/x.html

CONTAINER: Autbok... aus SDT: TCSYSTEM Tussis - Tastenbelegung

### Ergebnisse des Testlaufes vom 21.03.00 (06:28)

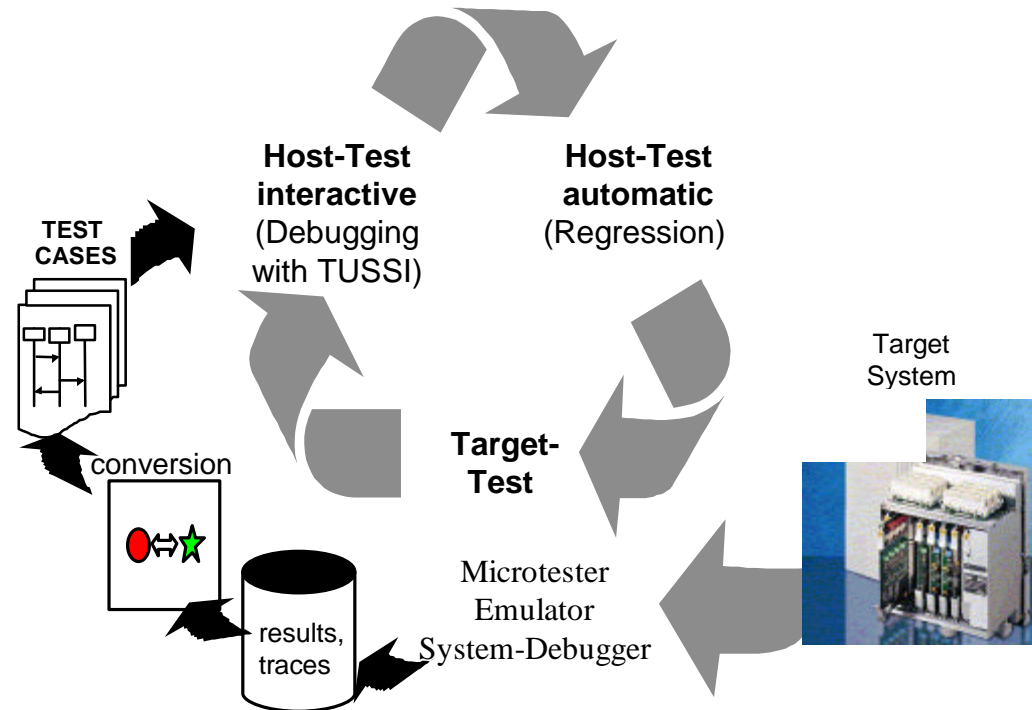
[zurück](#) [SDT-Homepage](#)

Eintrag	Beschreibung
NOT OK	Black-Box-Verhalten entspricht NICHT der Referenz
Warning	Black-Box-Verhalten OK, aber: Fehlereinträge gefunden
Warning	Black-Box-Verhalten OK, aber: Signal wurde verworfen
OK	Black-Box-Verhalten entspricht der Referenz
---	Auswertungsfehler: Toolfehler, Quelle oder Referenz fehlt o.ä. ...

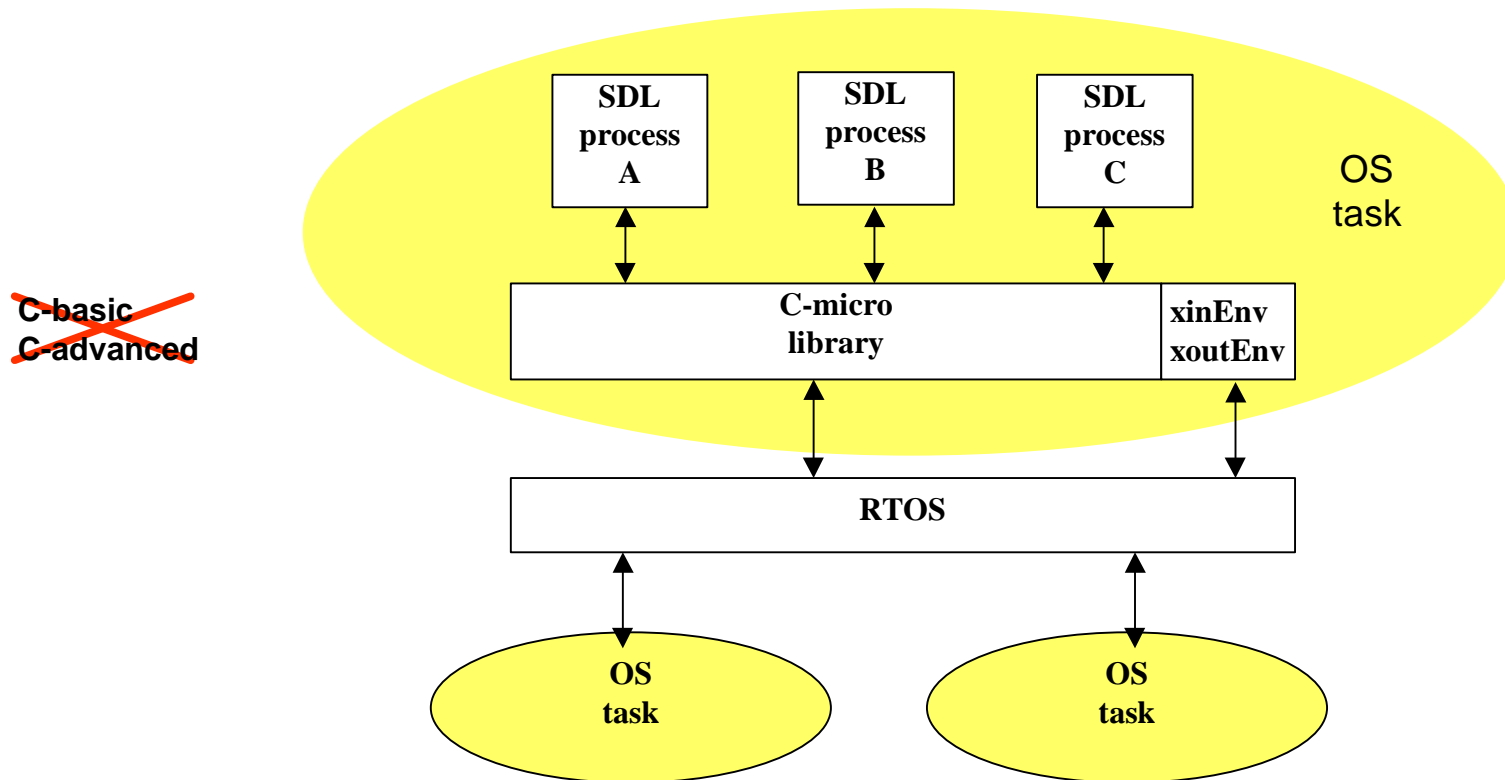
Um die Beschreibung der Testfälle zu sehen: >Klick< auf den Namen  
 Um die Quelle (.MPR) des Testfalls zu sehen: >Klick< auf (🟢 Allgemein) (🔴 HW-spezifisch)  
 Um das Simulator-log des Testfalls zu sehen: >Klick< auf (🔍)  
 Um das Input-file (.CMD) des Testfalls zu sehen: >Klick< auf (📄)

**HINWEIS:** Die interaktiven Features dieses Reports können derzeit nur unter *UNIX-NETSCAPE* genutzt werden !!!

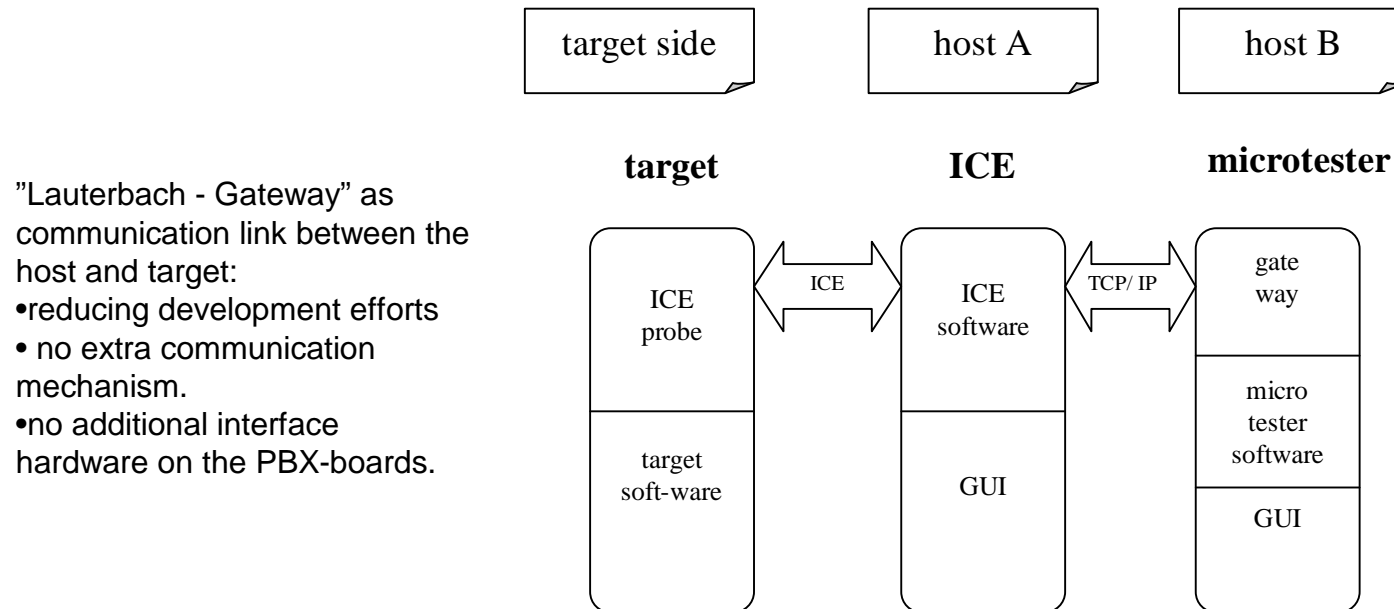
Testfall	Status	Verantw.	Vertretung
001 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_ACSEF01_ok	OK	Bartels	R.Endress
002 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_ACSEF02_ftt_ok	OK	Bartels	R.Endress
032 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_BCA12_Daten_senden_u_empfangen	OK	Horst Kozlik	S.Karg
033 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟡</a> CTI_BCA13_Daten_empf_m_Error	WARNING	Horst Kozlik	S.Karg
038 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_CPCA01	OK	Bartels	R.Endress
039 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_CPCA02_ok	OK	Kozlik	Bartels
040 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_CPCA03_timeout	OK	Bartels	R.Endress
041 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_CPCA04_invalid_connid	OK	Bartels	R.Endress
042 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_CHANGEEMON01_keine_Filter_ok	OK	Bartels	R.Endress
043 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🔴</a> CTI_CHANGEEMON02_filter_ok	*** NOT OK ***	Bartels	R.Endress
044 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🔴</a> CTI_CHANGEEMON03_alle_filter_ok	*** NOT OK ***	Bartels	R.Endress
045 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_CHANGEEMON04_timeout	OK	Bartels	R.Endress
070 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🟢</a> CTI_DIVERT10_invalid_callid	OK	Bartels	R.Endress
071 <a href="#">🔍</a> <a href="#">📄</a> <a href="#">🔵</a> CTI_DIVERT11_invalid_devid	NO REFERENCE	Bartels	R.Endress



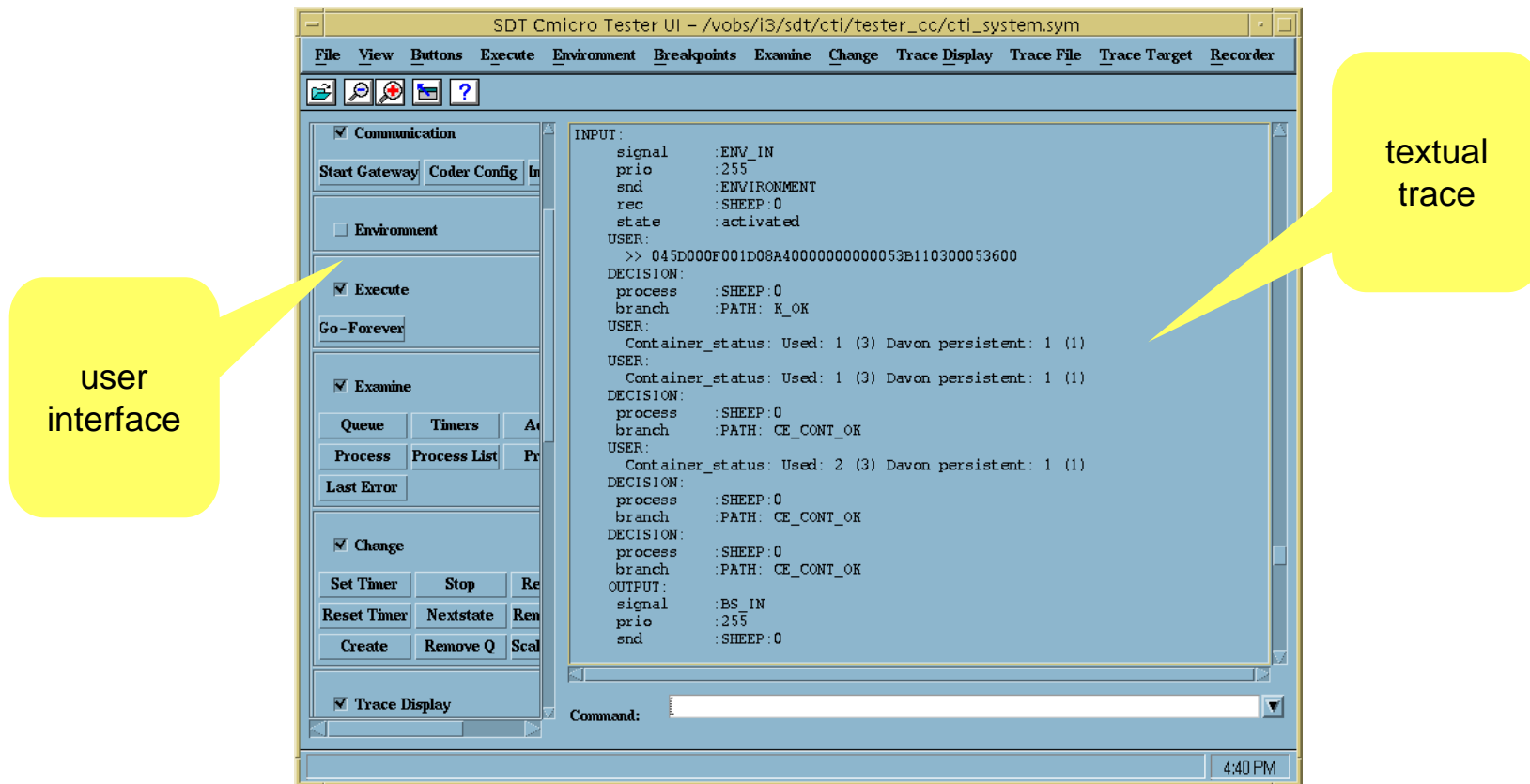
## “Light integration” of SDL-Systems into the RTOS



## Distribution of the microtester parts



## GUI of the microtester (Screenshot)



## **Advantages:**

- **microtester offers features for debugging usually only available in a development environment**
  - **setting breakpoints at SDL-level**
  - **drawing MSC-diagrams out of the target.**

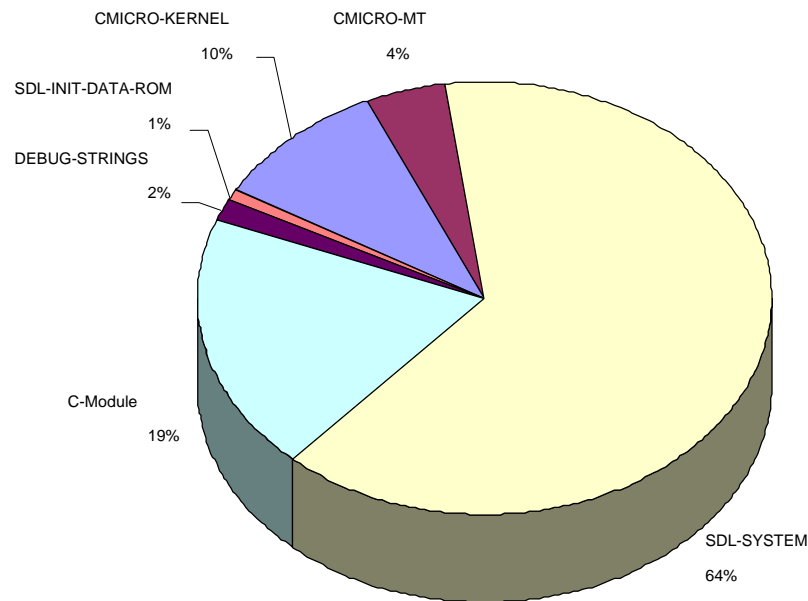


## Restrictions due to our target situation

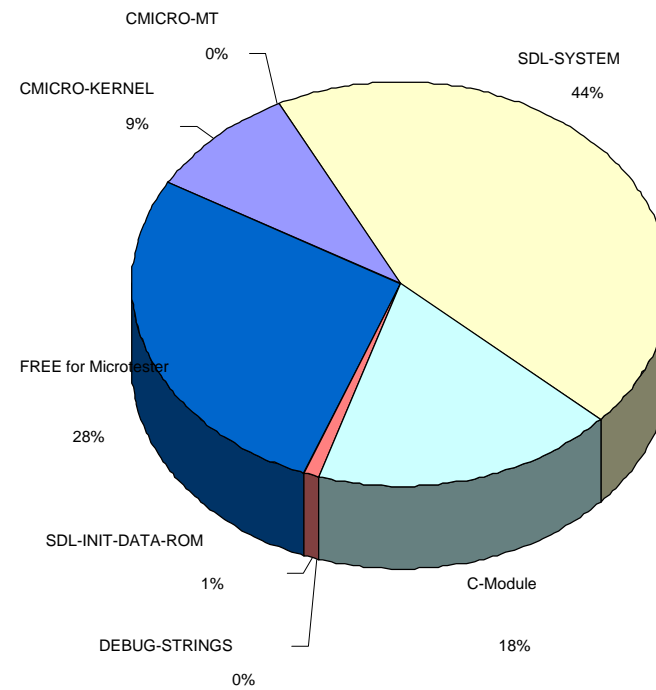
- graphical SDL trace causes a memory overflow.
- record and play mode cannot be used
  - ⇒ because of the multiprocessor system
  - ⇒ because not all tasks are designed with SDT
- recordings are always incomplete as not all events are visible to the microtester.
- different code generators and kernels on the host and target
- different scheduling between C-micro on the target and the C-basic on the host
  - ⇒ original task-loop of the C-micro-kernel has to be modified
- different data structures on host (SUN-Sparc) and target (Intel)
  - ⇒ different byte alignment and byte order leads to message coding (container)
- internal data structures of C-micro code and the C-basic code are completely different
  - ⇒ message coding & conditional compiling
- increased target software with microtester
  - ⇒ different mapping may have effects on debugging

## ROM

ROM SDL with the microtester



ROM SDL without microtester



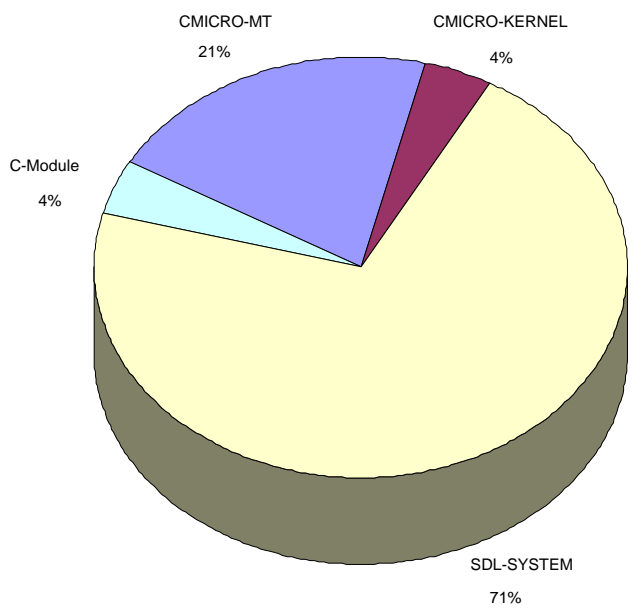
Development of SDL-Based Software - Practical Experiences

Presentation on SAM2000

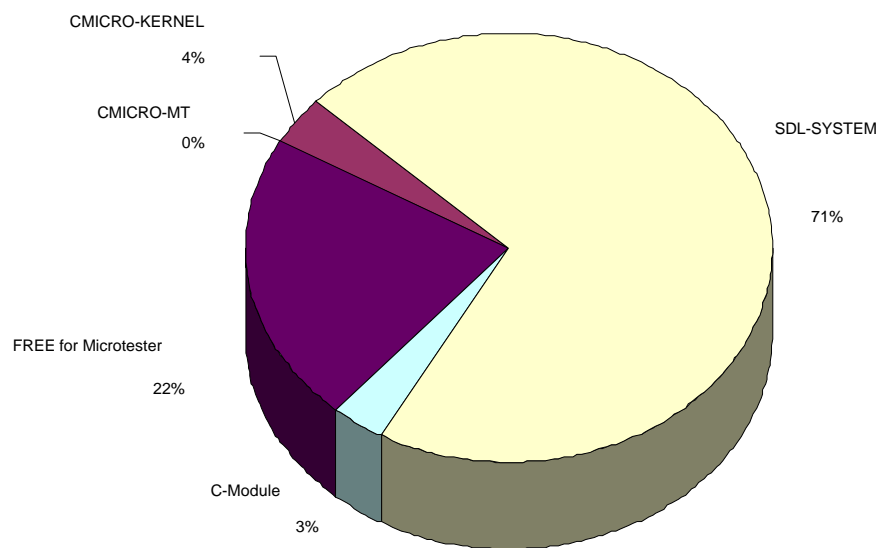
June 28<sup>th</sup> 2000

## RAM

RAM SDL with the microtester



RAM SDL without microtester

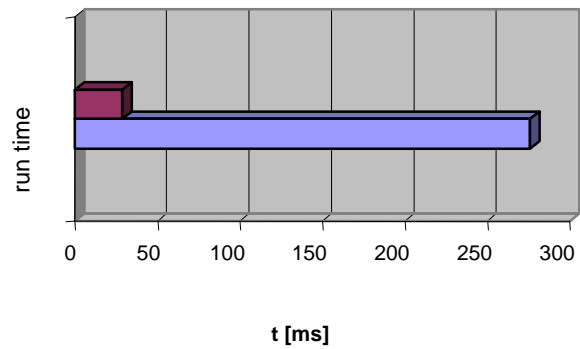


Development of SDL-Based Software - Practical Experiences

Presensation on SAM2000

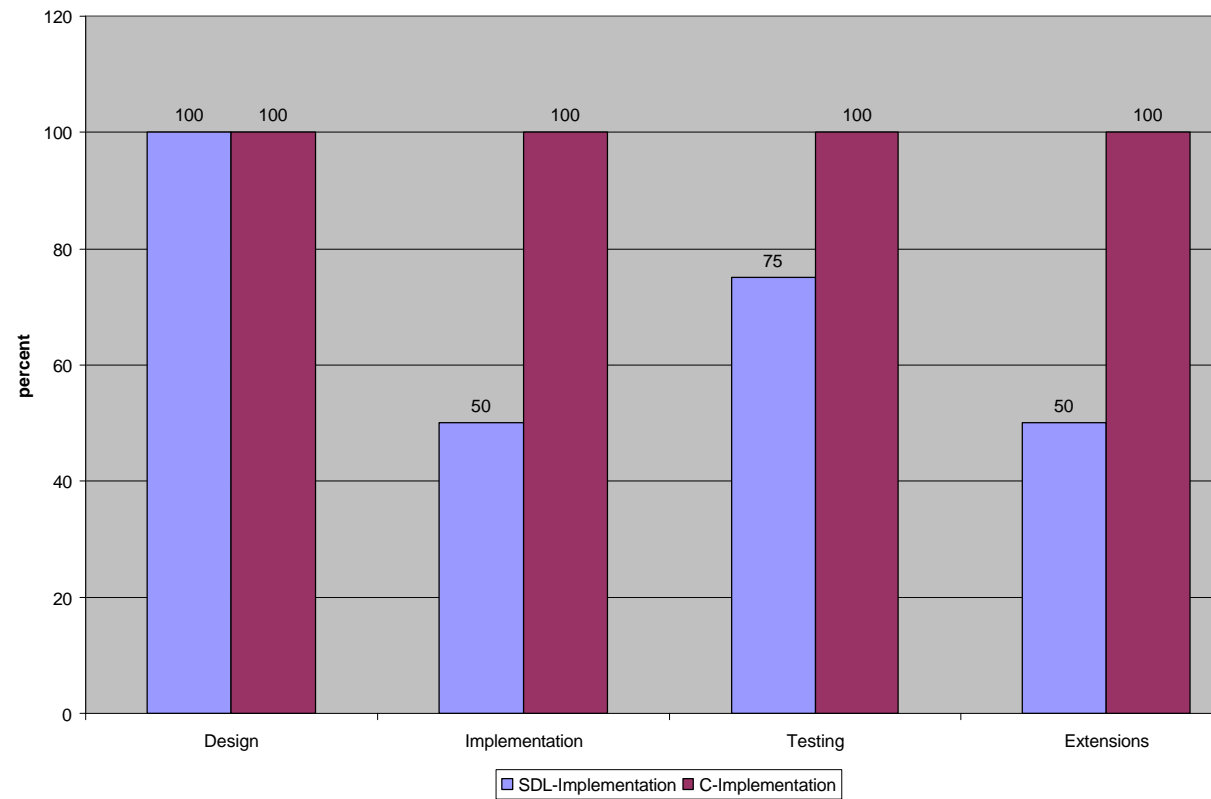
June 28<sup>th</sup> 2000

## Influence of the microtester on run time



	run time
■ without microtester	28,810
■ with microtester (best case)	275,980

Effort SDL contra C

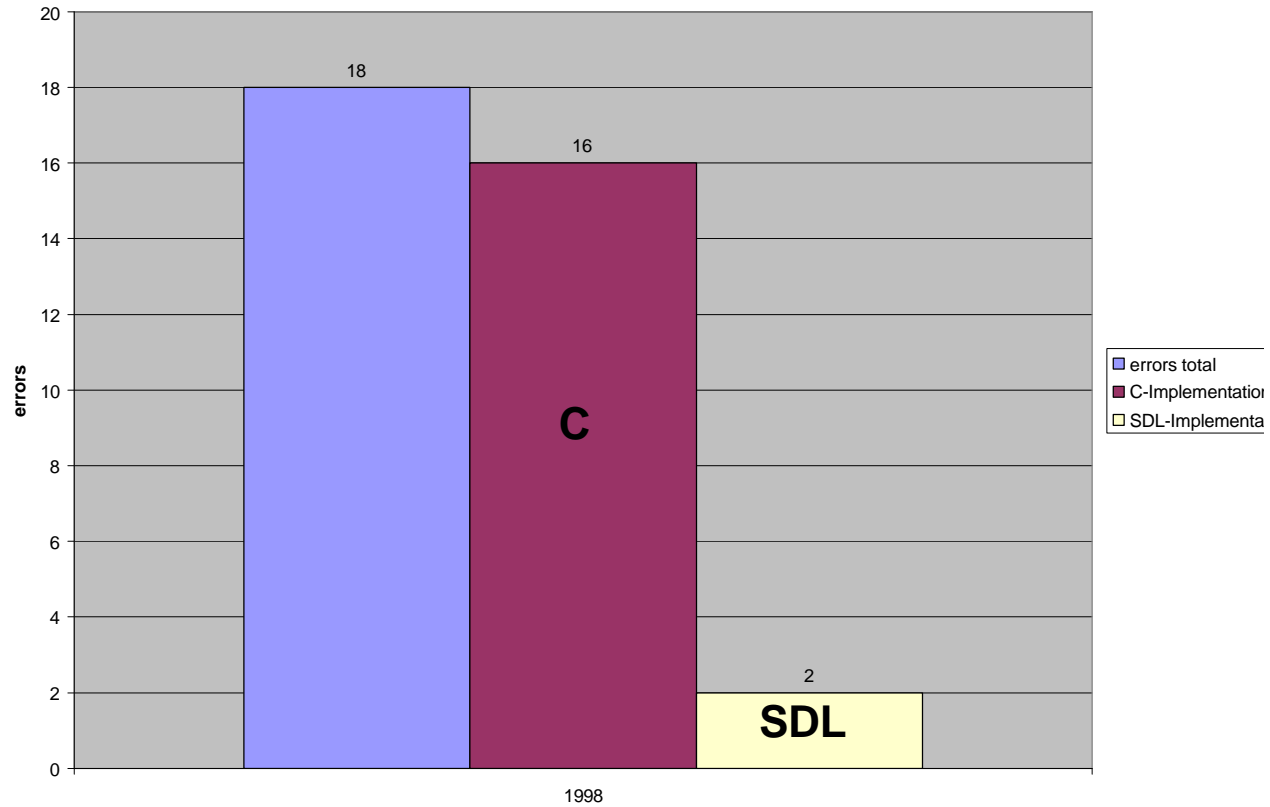


## Development of SDL-Based Software - Practical Experiences

Presentation on SAM2000

# Errors in SDL and C Implementations

Errors in SDL and C-Implementation



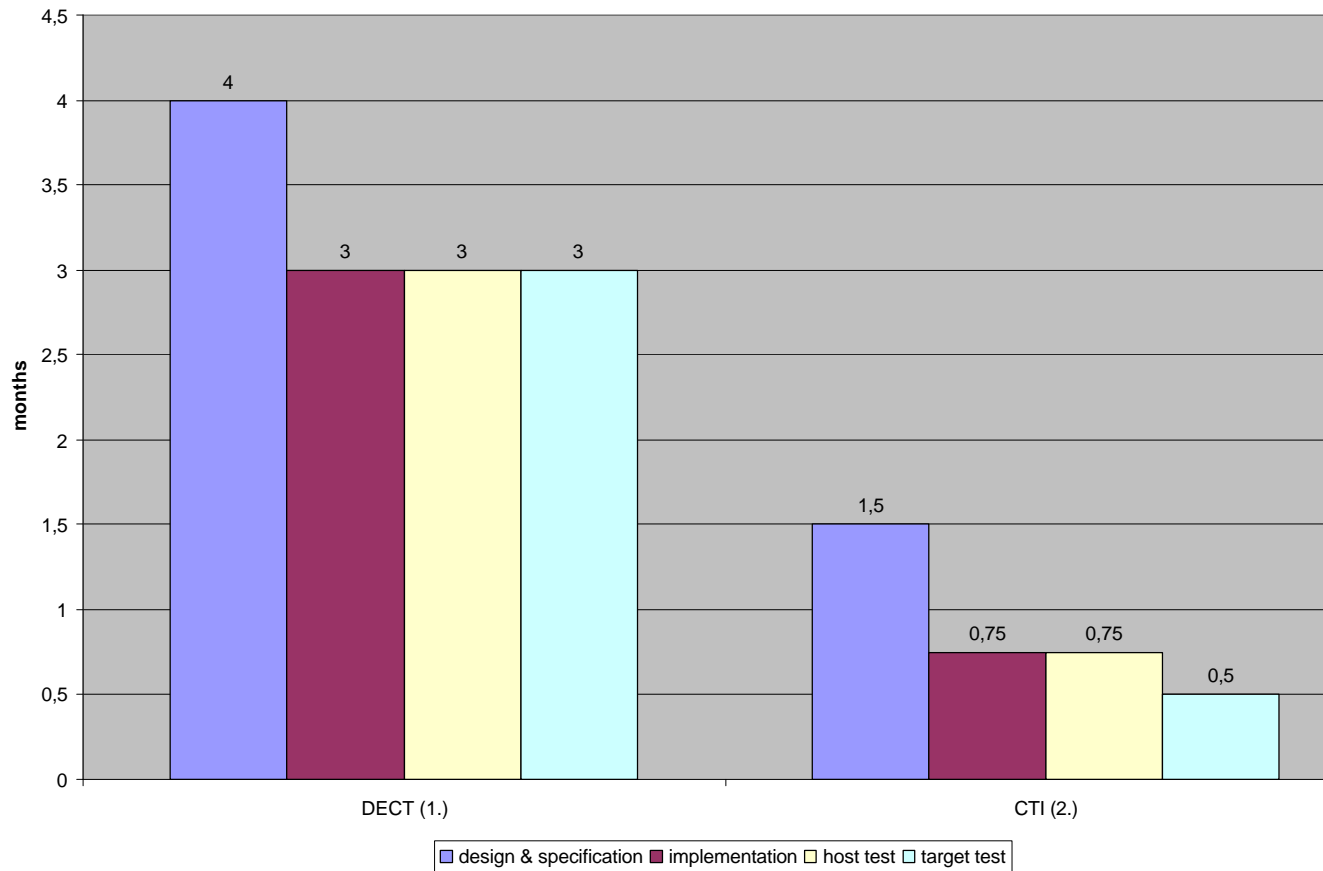
Development of SDL-Based Software - Practical Experiences

Presensation on SAM2000

# Development Speed with SDT



Development Speed, 1st and 2nd project with SDT



## Development of SDL-Based Software - Practical Experiences

Presensation on SAM2000

**++ Use SDL for protocol handling**

**++ Use SDL for state machines**

**-- Don't use SDL for software parts with loops**

**-- Don't use SDL for small data bases**



- **SDT tool chain + privat adaptations = a highly capable environment**
- **highly paralleled software development**
- **integration and regression testing throughout nearly the whole process**
- **automated documentation, testing and report generation**
- **measurable increase in development speed and product quality**

**Thank you  
for your attention.**

Josef Maier

Günther Kohler

Development of SDL-Based Software - Practical Experiences

Presensation on SAM2000