

# Software Development Methodologies

## Version Control 1

SOFTENG 750 2013-03-17

# Version Control Systems



<http://www.youtube.com/watch?v=5FEHlsRuWKI>

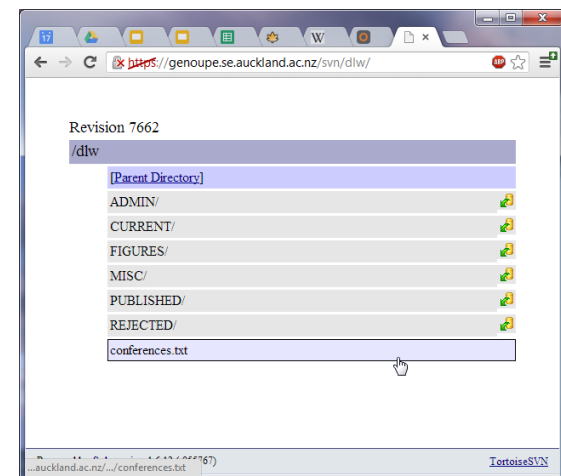
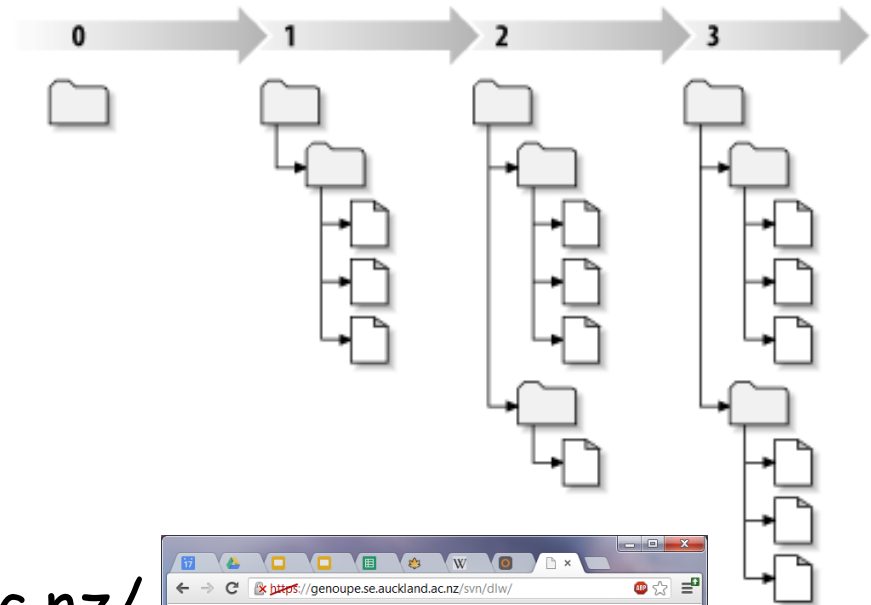
Visualization of activity on the Haiku git repository (2012) - Thanks to Justin Stressman

# Subversion - SVN (Recap)



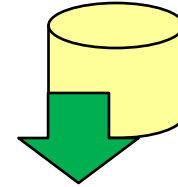
# Subversion (SVN)

- Centralized open-source VCS
- Supports **merging** (recommended) as well as **locking**
- **Complete file/folder structure** is versioned
- Revision names are sequential natural numbers (0, 1, 2, ...)
- Works with **HTTP server**:  
WebDAV/DeltaV protocol makes it possible to read repository with just a web browser:  
[https://subversion.sfac.auckland.ac.nz/svn/softeng750\\_2013\\_teamXY/](https://subversion.sfac.auckland.ac.nz/svn/softeng750_2013_teamXY/)



# Basic SVN Operations

- **Checkout:** create a working copy of a repository
  - Choose local folder for working copy
  - Enter the URL of the repository
  - Choose the revision to check out (HEAD revision is latest one)



- **Update:** update your working copy to the latest revision
  - If no newer revision exists: no effect
  - If you have changed your working copy: latest revision is automatically merged into your working copy
  - Textual merging conflicts have to be resolved manually



- **Commit:** write local changes to the repository
  - Fails if your local revision is out of date; update first
  - Creates a new revision on success

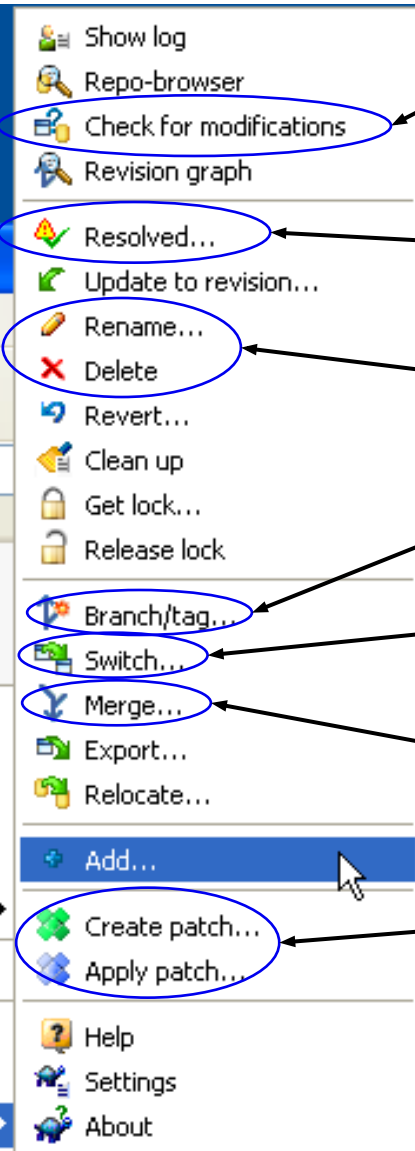
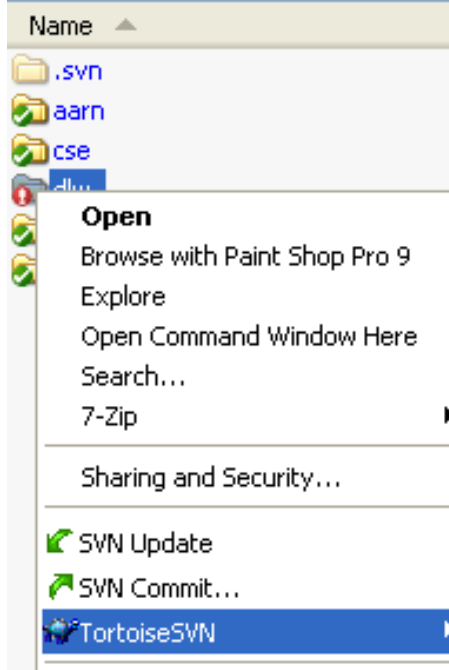


# TortoiseSVN



TortoiseSVN

✓ = Unmodified  
! = U changed  
it (needs commit)  
⚠ = Conflict  
🔒 = U have lock  
(release later)  
✖ = U deleted it  
+ = U added it



Check if somebody else has modified files or has acquired lock; also for stealing locks

Tell SVN that conflict in files has been resolved

Use these instead of normal ones!!! Also updates version info.

Create cheap copy of a folder.

Switch to the version in a cheap copy (like updating to it).

Merge revision range of branch into other branch.

Creating a file containing the local changes or use it to update working copy.

Right-click & drag

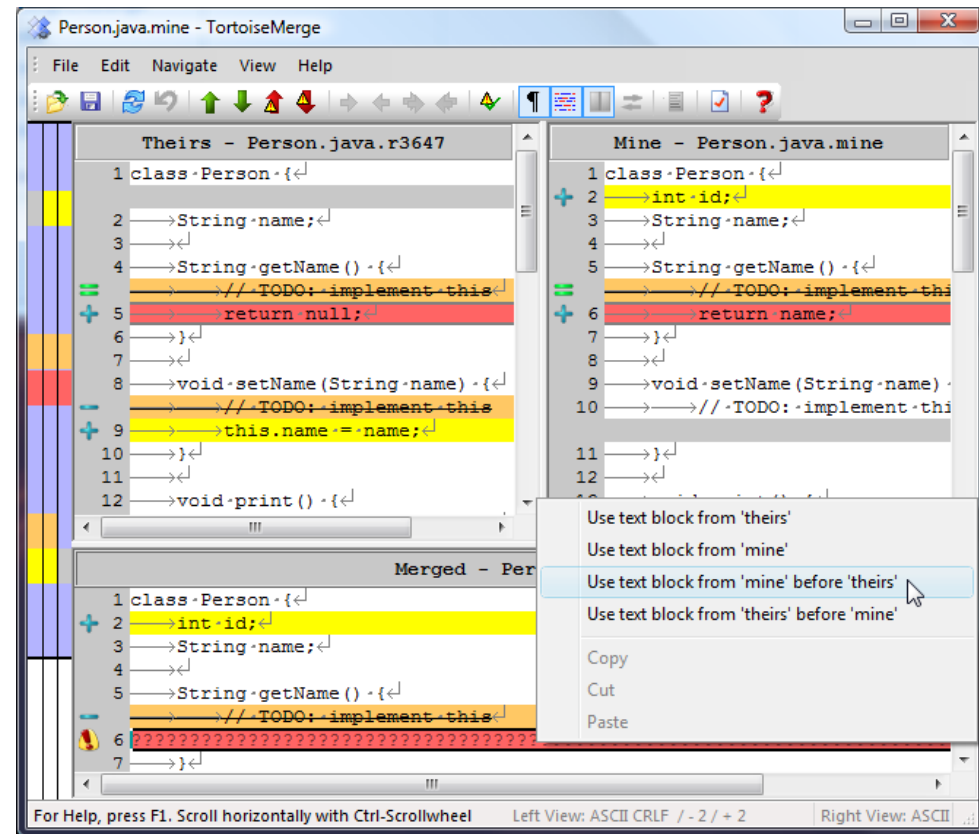
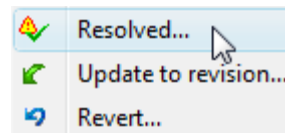
= copy/move & update version info

\*

# Resolving Conflicts

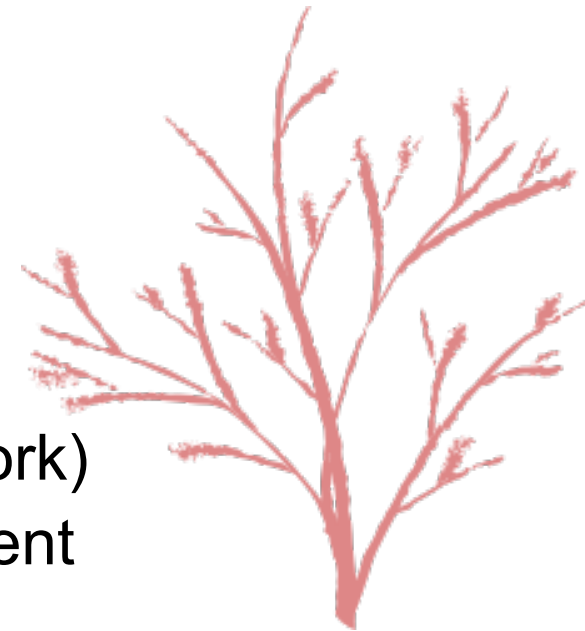


- After updating SVN might tell you someone committed a change that conflicts with your local changes
- Resolving the conflict means deciding how to merge the conflicting changes
- Supported by editor that shows conflicting changes and gives options to resolve it (e.g. use only one of two changes)
- When conflict is resolved, you must tell SVN



# Branches

- **Trunk / default / master branch**
  - Main line of development: everything converges there
- **Release branch**
  - Whenever you release software, create a release branch
  - Shortly before release, only bugfixes into release branch ("feature freeze")
  - Customer Feedback (e.g. bug reports) will relate to release branch
- **Experimental branch**
  - For unstable code (e.g. new features or rework)
  - To avoid interference with regular development
  - Merge into trunk once stable
- **Maintenance branch**
  - Created to maintain an old release





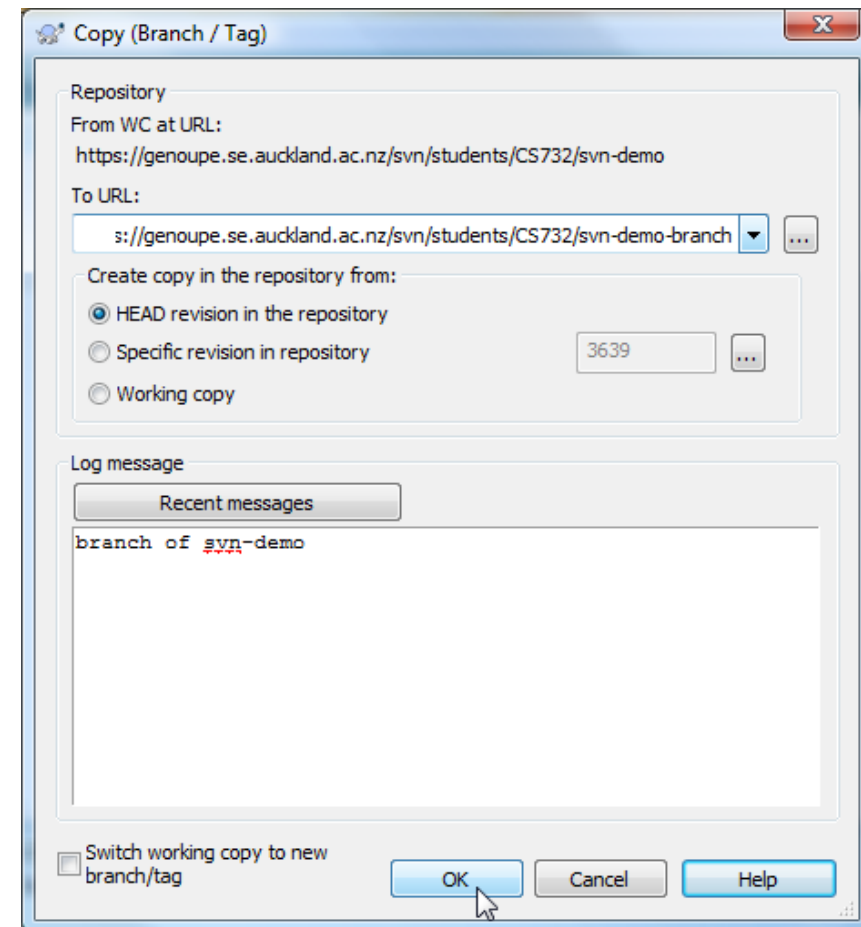
# Branching / Tagging



- Creates a copy of a folder in your repository
- **Branch:** the copy will be used for further development
- **Tag:** the copy is just for archival and will remain unchanged
- Usually three main folders: /trunk , /branches , /tags

How to do it:

1. Select folder to copy from  
(right-click on it, use menu)
2. In the dialog:  
select new folder to copy to
3. Select revision of that folder  
(usually HEAD)
4. Enter log message
5. Update parent folder of branch  
or tag to load it in the local working  
copy

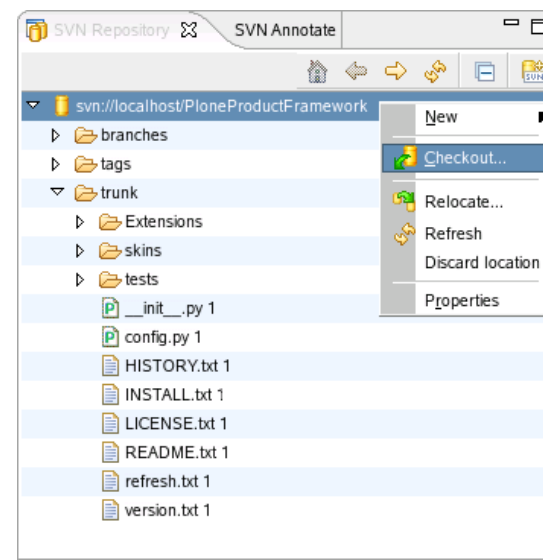
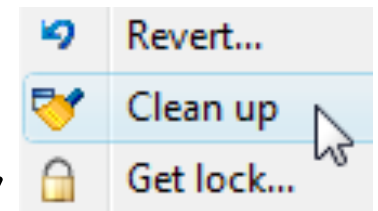
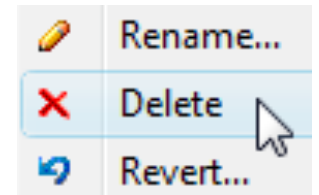




# Subversion Tips



- Don't forget to **add** your files/folders to the repo
- Delete and rename only using **SVN operations**
- If two SVN clients are running at the same time, there might be errors like "working copy locked"
- If something is wrong with working copy, use **cleanup** command
- If nothing else helps, delete local working copy check out a new one
- Various other clients available, e.g. Subclipse plugin for Eclipse



\*

# Version Control Best Practices



# 1. One Change at a Time

Complete **one change** at a time and commit it

- If you committing several changes together you cannot undo/redo them individually
- Sometimes individual changes are needed
- Sometimes individual changes need to be excluded



**Backup:** If you don't commit and your hard disk crashes...

- Your repository is your backup system
- Even if the repo is destroyed, other developers will probably have their own local copy

**Continuous integration** (see also XP practice)

- If you make several changes conflicts are much more likely
- Merging simple changes is much easier

## 2. Don't Break the Build

Only commit changes that preserve **system integrity**

- No "breaking changes" that make compilation or tests fail



**Productivity** (think of others)

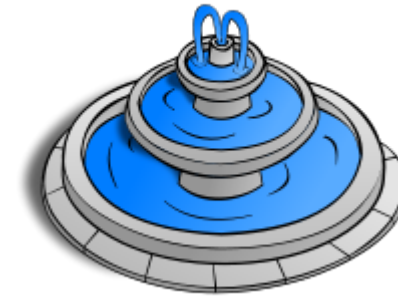
- All other developers will download your changes
- Any problem that was introduced will suddenly be everyone's problem

**Test-driven development** (see also XP practice)

- Write tests for every change
- Run tests before committing (at least some of them)

# 3. Only the Source

## Commit only source files



- I.e. files that are actually necessary for your software (including documentation)
- Not generated files (e.g. `.class`, `.exe`)
- Not temporary files (e.g. irrelevant data or log files)
- Source files are often textual and generated files binary
- Adding files should be a conscious decision!

## Why?

- Unnecessary files waste space  
(other people need to download them when checking out / updating)
- Most binary files are unmergeable  
(easily lead to conflicts that can't be resolved manually)

# 4. Use the Log

Write a log entry for each change

- What has been changed and why
- Like a short blog post (Twitter style or more)
- Easier to find good and bad changes



Revision	Time	Author	Description
4	1am	CodeCowboy	Added the files
5	1pm	CodeCowboy	More code
6	2pm	CodeCowboy	Minor change
Revision	Time	Author	Description
4	1am	CodeSheriff	Added files from our old repo at http...
5	1pm	CodeSheriff	Added Order.sort() for sorting OrderItems
6	2pm	CodeSheriff	Bugfix for #67: initialized variable

# 5. Communicate



Communicate with the other developers

## **Before changing existing code**

- See who else is working on it / has worked on it
- Ask that person about your change before committing (possibly show them a patch)

## **Before starting something new**

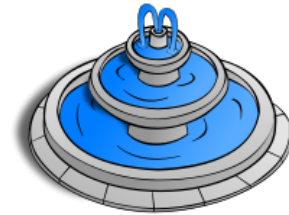
- Discuss with co-developers and agree on a design
- Make design proposal, point out design alternatives
- Always follow the project guidelines & specifications



# Version Control

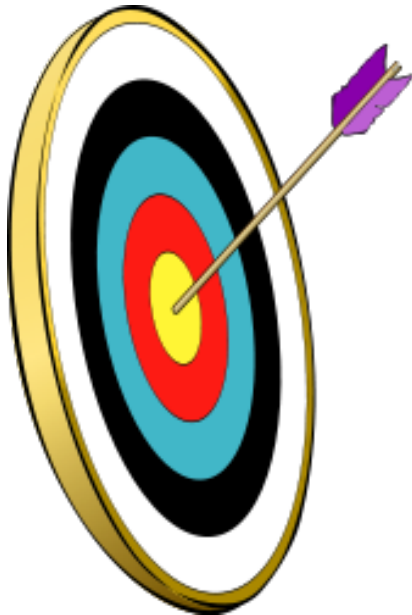
## Best Practices

1. Complete **one change at a time** and commit it
  - If you committing several changes together you cannot undo/redo them individually
  - If you don't commit and your hard disk crashes...
2. **Don't break the build**
  - Test your changes before committing
3. Commit **only the source** files (e.g. not `.class` files)
4. **Use the log** by writing a summary for each commit
  - What has been changed and why
5. **Communicate** with the other developers
  - See who else is working on a part before changing it
  - Discuss and agree on a design
  - Follow the project guidelines & specifications



\*

# Documenting Requirements

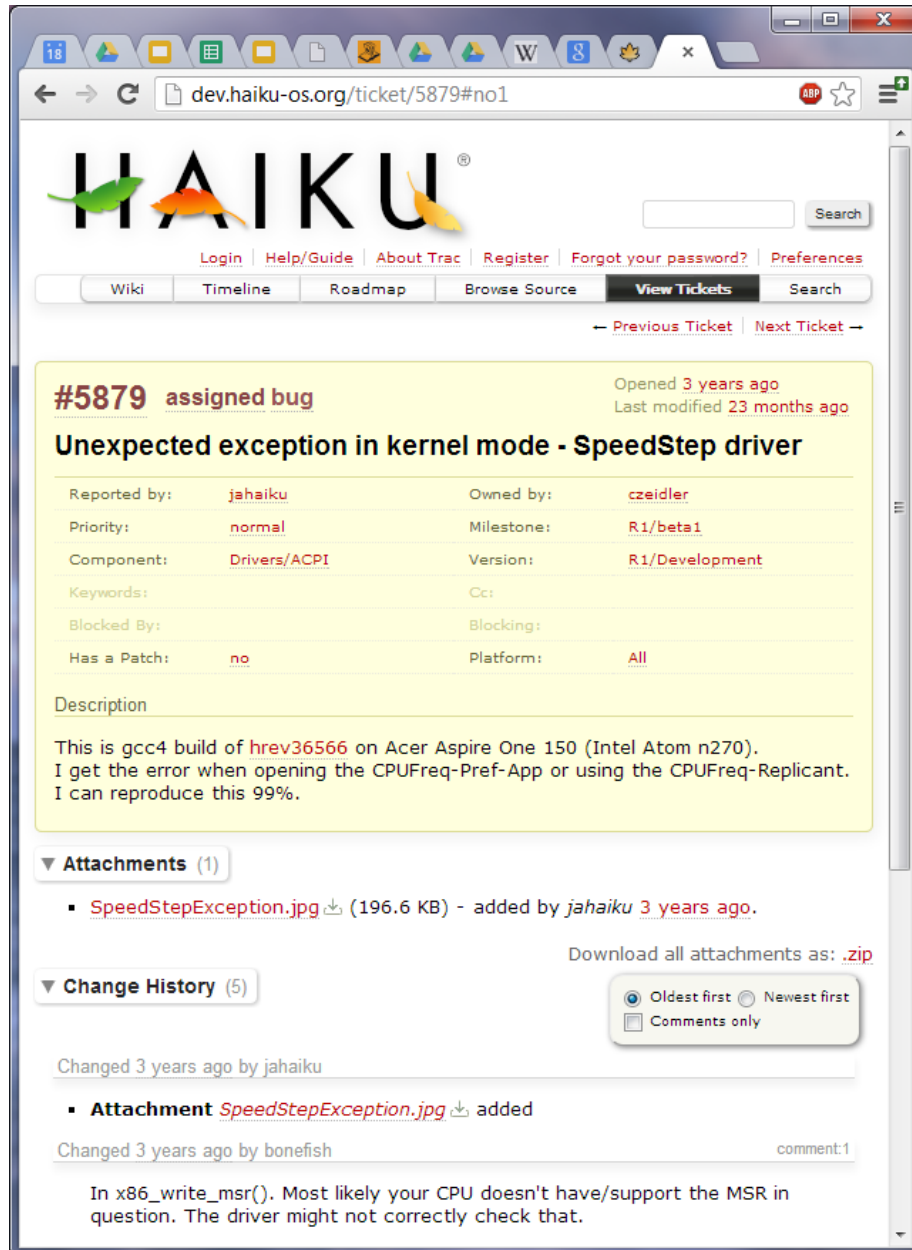


*...people don't know what they want and are willing to go through hell to get it.  
(Don Marquis)*

# Requirements Spreadsheet Example

	A	B	C	D	E	F	G
	Major Requirement	Minor Requirement	Task	Assignee	Estimate (h)	Actual (h)	Comments
1	Revise the GUI according to testing results		Replace textual buttons with graphical buttons and tooltips	G		2	Still need to review the buttons so that each button enables and disables appropriate buttons without making a conflict. Probably also need to tidy up the code and make it more understandable. I finally understand the reason that we have been taught to add descriptions and comments in the code at all time. It is so important in a group project and makes the other members to understand the code much eaier.
2						understand the code 1-2h, search for appropriate icons 1-2h, solving the button enabling and disabling problem 3-4h	
3			Regex filtering for state variables	O		10	
4			Regex filtering for OpenGL commands				
5	Enable managing of buffer state	Record buffers for each visible frame	Break after glSwapBuffers	H		10	Confused glSwapBuffers doesnt seem to exist O.o I've written a few lines in the shared doc on my thoughts. Please Critic and tell me if i missed something.
6			Read screen buffer				
7			Display screen buffer in GUI				
8			Store the screen buffer in PDStore				
9		Detect other buffers and read them					
10		Find other heuristics for detecting the end of a frame					
11	Support shaders	Represent shaders as ASTs	Create a parser	B		5	found glsl-unit: <a href="http://code.google.com/p/glsl-unit/">http://code.google.com/p/glsl-unit/</a>
12			Create PDStore AST model			~4h of googling, trying existing code. ~3h learning glsl-unit	
13			Extend parser to store shader into PDStore				
14		Intrument shaders (using glsl-unit)	Intercept shaders on compile call (glCompileShader~)	B		18	
15			Intrusement shaders with glsl-unit				~5h of play around with glsl-unit
16			Have probe load instrumented shader				
17			Readback debug data from instrumented shader				what debug data want to read?

# Using a Bug Tracker (Example: Trac)



dev.haiku-os.org/ticket/5879#no1

## #5879 assigned bug

Opened 3 years ago  
Last modified 23 months ago

### Unexpected exception in kernel mode - SpeedStep driver

Reported by: [jahaiku](#) Owned by: [czeidler](#)

Priority: [normal](#) Milestone: [R1/beta1](#)

Component: [Drivers/ACPI](#) Version: [R1/Development](#)

Keywords: Cc:

Blocked By: Blocking:

Has a Patch: [no](#) Platform: [All](#)

Description

This is gcc4 build of [hrev36566](#) on Acer Aspire One 150 (Intel Atom n270). I get the error when opening the CPUFreq-Pref-App or using the CPUFreq-Replicant. I can reproduce this 99%.

Attachments (1)

- [SpeedStepException.jpg](#) (196.6 KB) - added by [jahaiku](#) 3 years ago.

Download all attachments as: [.zip](#)

Change History (5)

Oldest first  Newest first   
Comments only

Changed 3 years ago by [jahaiku](#)

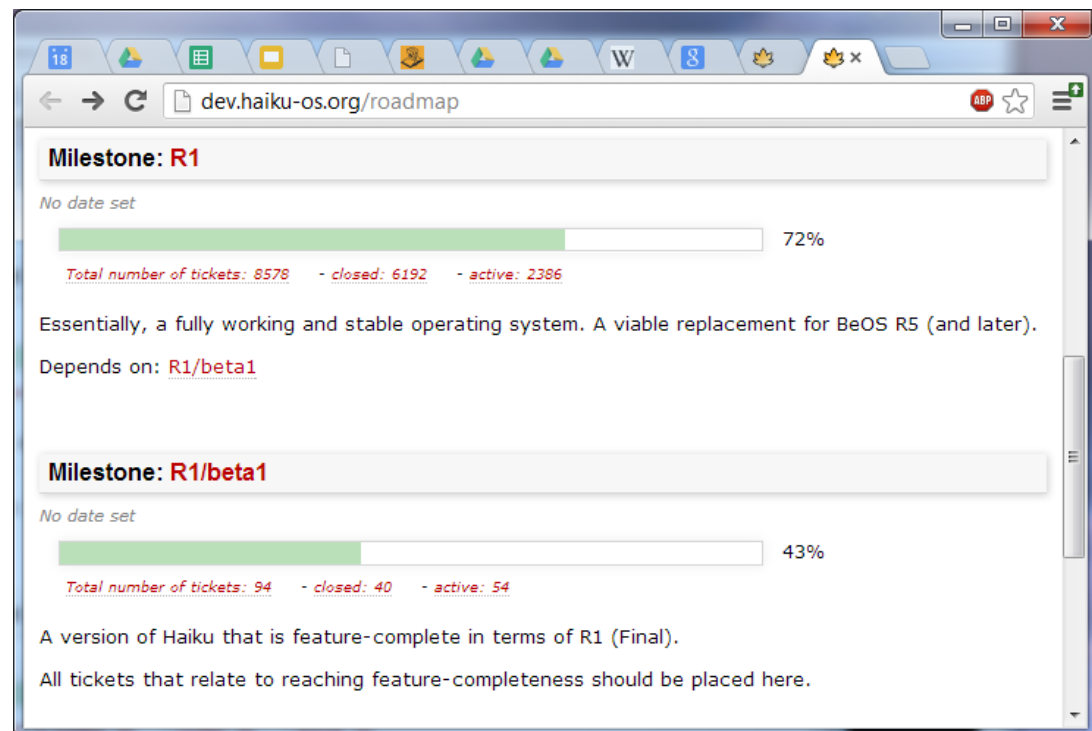
- Attachment [SpeedStepException.jpg](#) added

Changed 3 years ago by [bonefish](#) comment:1

In `x86_write_msr()`. Most likely your CPU doesn't have/support the MSR in question. The driver might not correctly check that.

Status: assigned (14 matches)

Ticket	Summary	Owner	Type	Priority	Component
#8345	PANIC: ASSERT FAILED ... x86/paging/pae/x86VMTranslationMapPAE.cpp:231	bonefish	bug	critical	System/Kernel
#6824	ShowImage crashes after browsing thru many photos	axeld	bug	high	Applications/ShowImage
#5179	Touchpad buttons not working correctly due to missing ClickPad support	siarzhuk	enhancement	normal	Drivers/Mouse/PS2
#5474	[kernel] PANIC: got an in use page 0x8107c040 from the free pages list	mmlr	bug	normal	System/Kernel
#5777	Kernel starts paging when writing to slow media	axeld	bug	normal	System/Kernel
#5879	Unexpected exception in kernel mode - SpeedStep driver	czeidler	bug	normal	Drivers/ACPI
#6882	BTextControl layout issues	yourpalal	bug	normal	Kits/Interface Kit
#7008	Implement missing <code>&lt;search.h&gt;</code> , <code>&lt;stdio.h&gt;</code> , <code>&lt;stdlib.h&gt;</code> , <code>&lt;wchar.h&gt;</code> functions and definitions	bonefish	bug	normal	System/POSIX
#7078	Allowing shortcuts without the <code>B_COMMAND_KEY</code> modifier	leavengood	enhancement	normal	Kits/Interface Kit
#8275	Deadlocking apps when opening documents	leavengood	bug	normal	System/runtime_loader
#8306	[PATCH] Allow entire tree to be compiled with <code>DEBUG=2</code>	umccullough	bug	normal	- General
#8435	[installoptionalpackage] error when installing OpenSSL	mmadia	bug	normal	Applications/Command Line Tools



dev.haiku-os.org/roadmap

### Milestone: R1

No date set

72%

Total number of tickets: 8578 - closed: 6192 - active: 2386

Essentially, a fully working and stable operating system. A viable replacement for BeOS R5 (and later).

Depends on: [R1/beta1](#)

### Milestone: R1/beta1

No date set

43%

Total number of tickets: 94 - closed: 40 - active: 54

A version of Haiku that is feature-complete in terms of R1 (Final).

All tickets that relate to reaching feature-completeness should be placed here.



# Today's Summary

1. Always use a VCS when doing software development
2. Stick to best practices to get the most out of your VCS (and have a good team experience)
3. Document and manage the product requirements (your customer will love you for it)

## **Milestone 3 (Deadline: Lab on Thursday)**

1. Requirements breakdown (product backlog)
2. First iteration plan (sprint backlog)  
with task breakdown, allocation & estimates

# Quiz

1. What are the steps of working with a SVN repository?
2. When should a branch be created and how?
3. Explain 3 best practices of version control and describe what could happen if they are not followed.
4. How would you document product requirements?

```
#include <stdio.h>
#define c(C) printf("%c",C)
#define C(c) ((int*)(C[1]+8))[c]
main(int c, char
%C[1]
1)+2 )[]= c[52]*c(\
'C'+ '4'/4 )/c*c
=0; c<48; ++ c)/c*c
'C' C[1][c] -c) =
0; C[0]<8; { ** C
)++ )C[1][c- 'c'] =
*(C[ 1]+c*'c')+ C[1][
88- c]+(C[1 ]]**C
48*c +88)=32 }; [
%C[4]=*C[2]= 75 ?
*((C[2]+3)-2 )=-707
1:0:0;C(0)=C( 1)w=0
}while(*C[2]? C[2][1]
?*(C[2]+2)?:1 :0:0:0)
{&C[ *C[ 2 ]>'w'};
C[1]=0;C[1] [2]++;*C
[2]=0;while C[1]+*C[
2]=0872+( C[2][3]&&
*(C[2]+3)< 'w')*C[2]
==a?*(C[ 2][1]==48):
%C[2]>88 ?3-*(C[2]==\
'm??:0 )>:;C(0)=C(1)>
C(0)?C(1 ):C(0);c=3;*
(C+2)+3;printf(" %d\
 %d\n", 56+5*( 0),80** (C[2] ++))
;C[2]=0 ;C[2] **c)*C[2] =0;
while(C[3] [1,- 1]--)? c*c( **
0; ** C[2] 80; (** C[4]) C
[2] ** C[2]; *C[3] ++
=0; *C[ 3] ==*C== 51;)* *C
18 ;)* *C 88;=270 :288 ;c[2
-1 ])* ( *C [2]);c*c( *(C[
1] =0;[ *(C[ 1]42; *(C[ 1] ++)*C
[3 ] )[[ *(C[ 4]?**C>2866* *C42 ?C[1]
42 +** C+1) -3***C:0; **C== 1266*
 *C <847 <C[ 1 ] 7***C+87 +( *C[ 1 ] 1);
0 )?c( *C[3 ]++) ;c*c(C [1]=0; [C]
2) =C[1 ]<C( 0); ( *(C [2]=C [2]
1) -48; c=*( C[2]4= 82); c=*( C[ 0]
-4 * *C[ 3] 0 )>=505- C[2][ c] -7)* *C
[2]=< *C' -18*( C[2][ c] <77)-42*(
 *C[ 4 ] 1-7* (C[2] [c]<'C' ))-8;c*c(C
3)=0;
C[3]< *(C[2]?*C[2
]>'a' *C[ 2] 7C[ 2][1]:12
42*( *C [2]> 84) +2-(C[2 ] [3])
*(88 == *( *C [ 2] 42) [0]) -
(C[2] [0 ] ==*m' ) )>C(2)?C(0 )>C(
2):0 )>C(3)++ C[1]+c(C [1][4] | * C[2]&&
 *C[2 ]<'a'?*C[2] ==87C[3] ==0;7**C >2766**C
<51 ?C[ 2] [1] ==8738; C[2][ 1 ]==879:1:
0:0 :83 >* C[2 ]?c<764 c>-97C[1 ]|( *C[2]<
487 'a' *5 -42' ?>*C[ 2]< 817870 :171)+
3*c 1: 0) *C [2]>'a' 7***C>2866**C429 661
 *C[ 1 ] 2] |[( *C[ 2] ==166*( *C[ 0] ) [0]
<34 66 31 <7* C[0]C[3] <2715+228*C[3 ] :0:1
 *C[ 2] == 37C [3]<2 66**C>2866**C <4866C
[3]< 27 C[ 1] 7* * *C[4]514C [3]:0:1
7== *C [ 2] [2 ]&& *C>2866**C <4266C[3
]>27C [1 ] 17 ***C <1354C[3] )> *C[2]
<'a'? [c >2 66 c<37C[3
]>27C [1] |[( *C) [2]<27
207:205)+7* c=C[3 ] :1:C[
2][2]==48 66 (c==2|| c==
1- 2*( *C[3]2 )?7
98 :0:0)|( *C) [2
] ?
C[ ] 7C[3]< 2
66 *C[3]> 7 66
c< 166c >-24 75:0: * C
[3 ]<666c >1 66c<24 7 *
|C[3] <2 66( ** C
==88 66 *C[3 ]>24 |)* C[ 3
]>22 66 88 ***C[ ] *
C[3 ]< 2 66**C=10 ) 7
5* 51: 0) | (7== (
 *C[ ] 2]?*C[3] <
87c>1366 c<2366C[3]<27C
[1]1444+ 7*c(C[3]):0:C[
3] 66 c<-1466c>- 247C[
1] 7 *c <400+C[3 ]:0:1
0) :1 C (3)?**C> 2166C
[0 ] | 0] <32
?C [ 1 ] |[(
 * C ) [ 2 ] +323
+7 ** *C :1:38
+ 1 < ** C66 '0'>
 * *C ? C[1]
C[ 2] [2]+82
+ 7* **C):0:0:0);C[1
] [ 3] ++;C[2]=3;c(0)
;C [ 3 ]==2;)*C[3]=0;}
```

International Obfuscated C Code Contest (ioccc.org) - deckmyn 2012

A music notation program. A music font is encoded in the whitespaces of the code.