A Tcl-based Kanban board for Fossil tickets

Steve Landers
steve@digitalsmarties.com

WARNING
OPINIONS AHEAD

Steve’s Opinions

- software project management is hard!
- one size fits nobody
- the means often becomes the end

Why Fossil Tickets?

- OSS solutions clunky
- hosted alternatives are OK but …
- DVCS integration
- customisation

Fossil tickets

The good

- distributed, integrated, flexible

Fossil tickets

The good

- distributed, integrated, flexible

The bad

- lack of process

Fossil tickets

The good

- distributed, integrated, flexible

The bad

- lack of process

The ugly

- the interface

Kanban

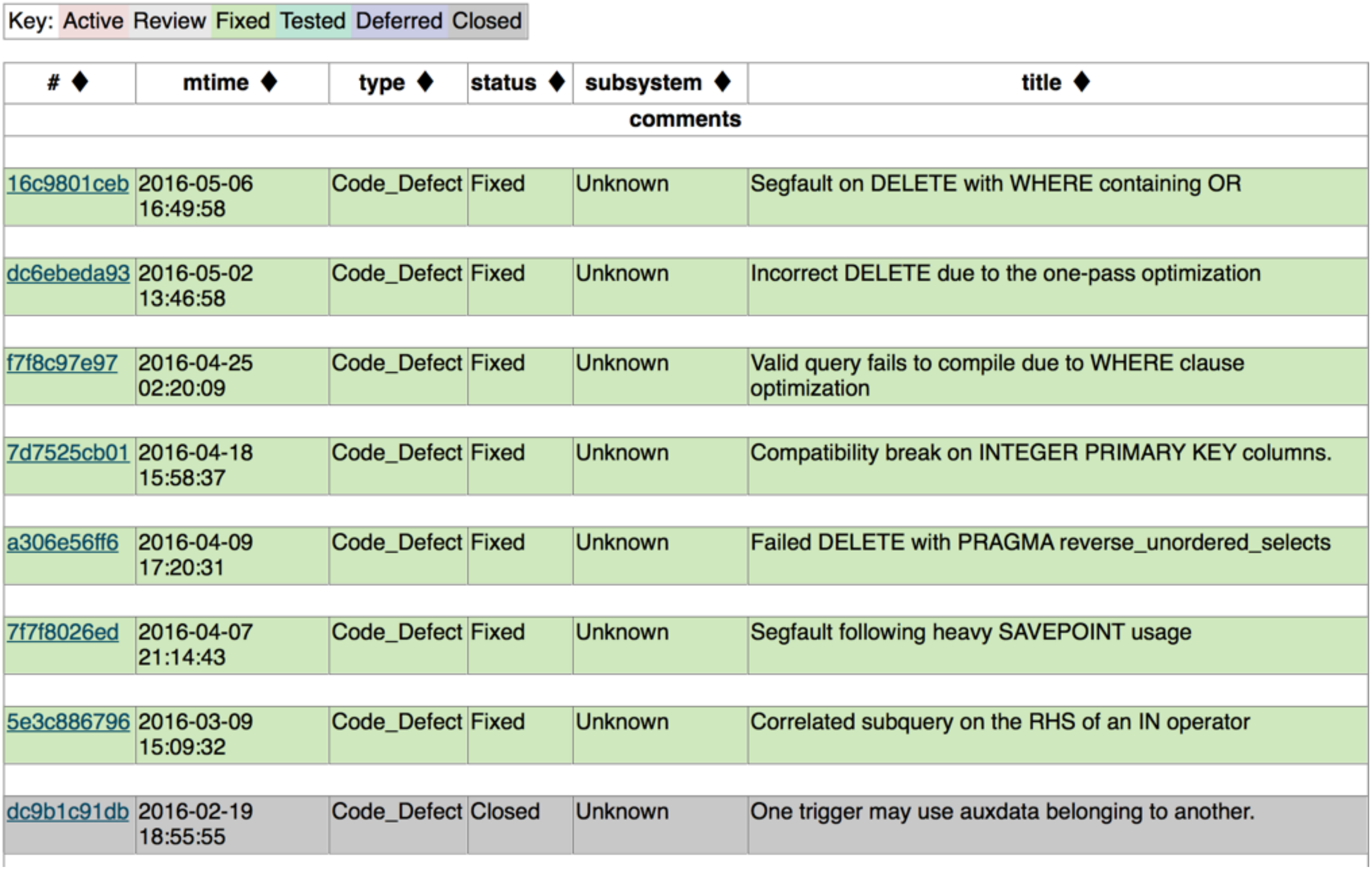
- Japanese for “signboard” or “billboard”
- Toyota1950s
- visual planning
- continuous delivery
- adopted by “agile” advocates

Nomenclature

- projects
- tasks
- activities

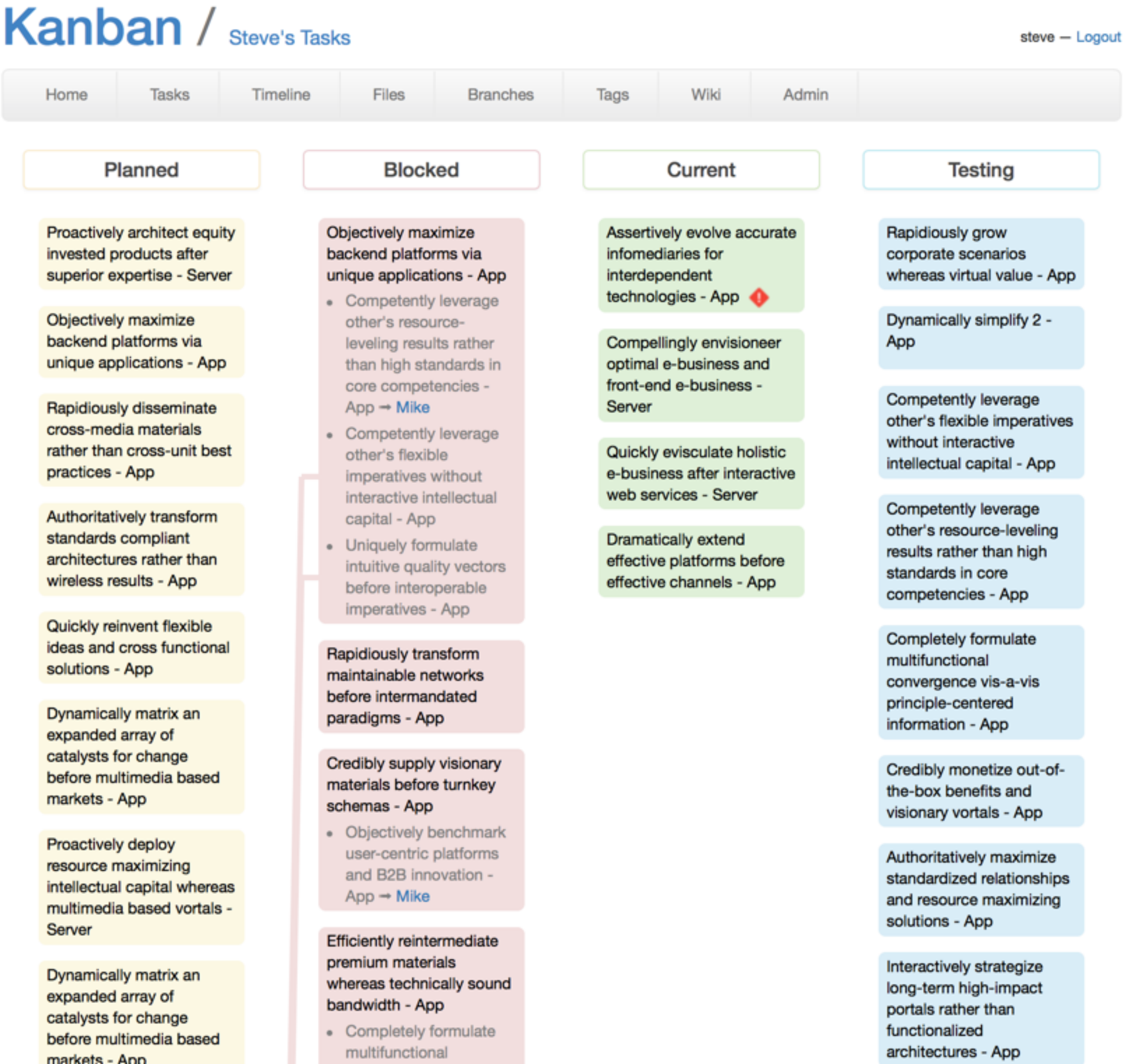
Tcl Kanban

- html / js / css page per user or per-project
- generated by Tcl script
- uses Bootstrap + jQuery + jQuery UI



Key: Active Review Fixed Tested Deferred Closed

# ♦	mtime ♦	type ♦	status ♦	subsystem ♦	title ♦
comments					
16c9801ceb	2016-05-06 16:49:58	Code_Defect	Fixed	Unknown	Segfault on DELETE with WHERE containing OR
dc6ebeda93	2016-05-02 13:46:58	Code_Defect	Fixed	Unknown	Incorrect DELETE due to the one-pass optimization
f7f8c97e97	2016-04-25 02:20:09	Code_Defect	Fixed	Unknown	Valid query fails to compile due to WHERE clause optimization
7d7525cb01	2016-04-18 15:58:37	Code_Defect	Fixed	Unknown	Compatibility break on INTEGER PRIMARY KEY columns.
a306e56ff6	2016-04-09 17:20:31	Code_Defect	Fixed	Unknown	Failed DELETE with PRAGMA reverse_unordered_selects
7f7f8026ed	2016-04-07 21:14:43	Code_Defect	Fixed	Unknown	Segfault following heavy SAVEPOINT usage
5e3c886796	2016-03-09 15:09:32	Code_Defect	Fixed	Unknown	Correlated subquery on the RHS of an IN operator
dc9b1c91db	2016-02-19 18:55:55	Code_Defect	Closed	Unknown	One trigger may use auxdata belonging to another.



Kanban / Steve's Tasks

Planned

Proactively architect equity invested products after superior expertise - Server

Objectively maximize backend platforms via unique applications - App

Rapidiously disseminate cross-media materials rather than cross-unit best practices - App

Authoritatively transform standards compliant architectures rather than wireless results - App

Quickly reinvent flexible ideas and cross functional solutions - App

Dynamically matrix an expanded array of catalysts for change before multimedia based markets - App

Proactively deploy resource maximizing intellectual capital whereas multimedia based vortals - Server

Dynamically matrix an expanded array of catalysts for change before multimedia based markets - App

Blocked

Objectively maximize backend platforms via unique applications - App

- Competently leverage other's resource-leveling results rather than high standards in core competencies - App → Mike
- Competently leverage other's flexible imperatives without interactive intellectual capital - App
- Uniquely formulate intuitive quality vectors before interoperable imperatives - App

Rapidiously transform maintainable networks before intermandated paradigms - App

Credibly supply visionary materials before turnkey schemas - App

- Objectively benchmark user-centric platforms and B2B innovation - App → Mike

Efficiently reintermediate premium materials whereas technically sound bandwidth - App

- Completely formulate multifunctional

Current

Assertively evolve accurate infomediaries for interdependent technologies - App ❗

Compellingly envisioneer optimal e-business and front-end e-business - Server

Quickly evisculate holistic e-business after interactive web services - Server

Dramatically extend effective platforms before effective channels - App

Testing

Rapidiously grow corporate scenarios whereas virtual value - App

Dynamically simplify 2 - App

Competently leverage other's flexible imperatives without interactive intellectual capital - App

Competently leverage other's resource-leveling results rather than high standards in core competencies - App

Completely formulate multifunctional convergence vis-a-vis principle-centered information - App

Credibly monetize out-of-the-box benefits and visionary vortals - App

Authoritatively maximize standardized relationships and resource maximizing solutions - App

Interactively strategize long-term high-impact portals rather than functionalized architectures - App

Cosmetic changes to Fossil

- new columns in ticket schema
- change some defaults to match
- edit HTML templates to match
- ticket change TH1 hook
- update reports page + report definitions
- create new branch called ‘status’

```
CREATE TABLE ticket(
  …
  type TEXT,
  status TEXT,
  subsystem TEXT,
  priority TEXT,
  severity TEXT,
  foundin TEXT,
  private_contact TEXT,
  resolution TEXT,
  title TEXT,
  comment TEXT,
  created TEXT,
  assigned TEXT,
  effort TEXT,
  blocked TEXT
);
```

```
set subsystem_choices {
  App
  Dev
  Marketing
  Server
  Website
}

set created_choices {
  Steve
  Mike
}

set assigned_choices {
  Steve
  Mike
}
```

```
set type_choices {
  Action
  Problem
}

set priority_choices {
   Normal
   High
   Low
}
set effort_choices {
   Hours
   Days
   Weeks
   Months
}
```

```
set status_choices {
  Unscheduled
  Planned
  Current
  Testing
  Completed
  Archived
}
```

Modify HTML pages

- New ticket
- View ticket
- Edit ticket

Generating the board

- Fossil / settings / enable tcl
- Fossil / settings / Edit change script

```
tclInvoke exec tclsh ~/bin/kanban.tcl $assigned
```

- generates per-user HTML/CSS/JS file

Sharing the board

- copy to web server

Sharing the board

- commit to status branch
- hide status branch from timeline
- access via Fossil Project Documentation

```
doc/status/$name.html?login=$login

doc/status/steve.html?login=stevel
```

Demonstration

Notification

- Slack interface
  - write to @person or #channel
  - notify as needed

```
fossil new app task Prepare Presentation
created by Steve now current
```

Where to now?

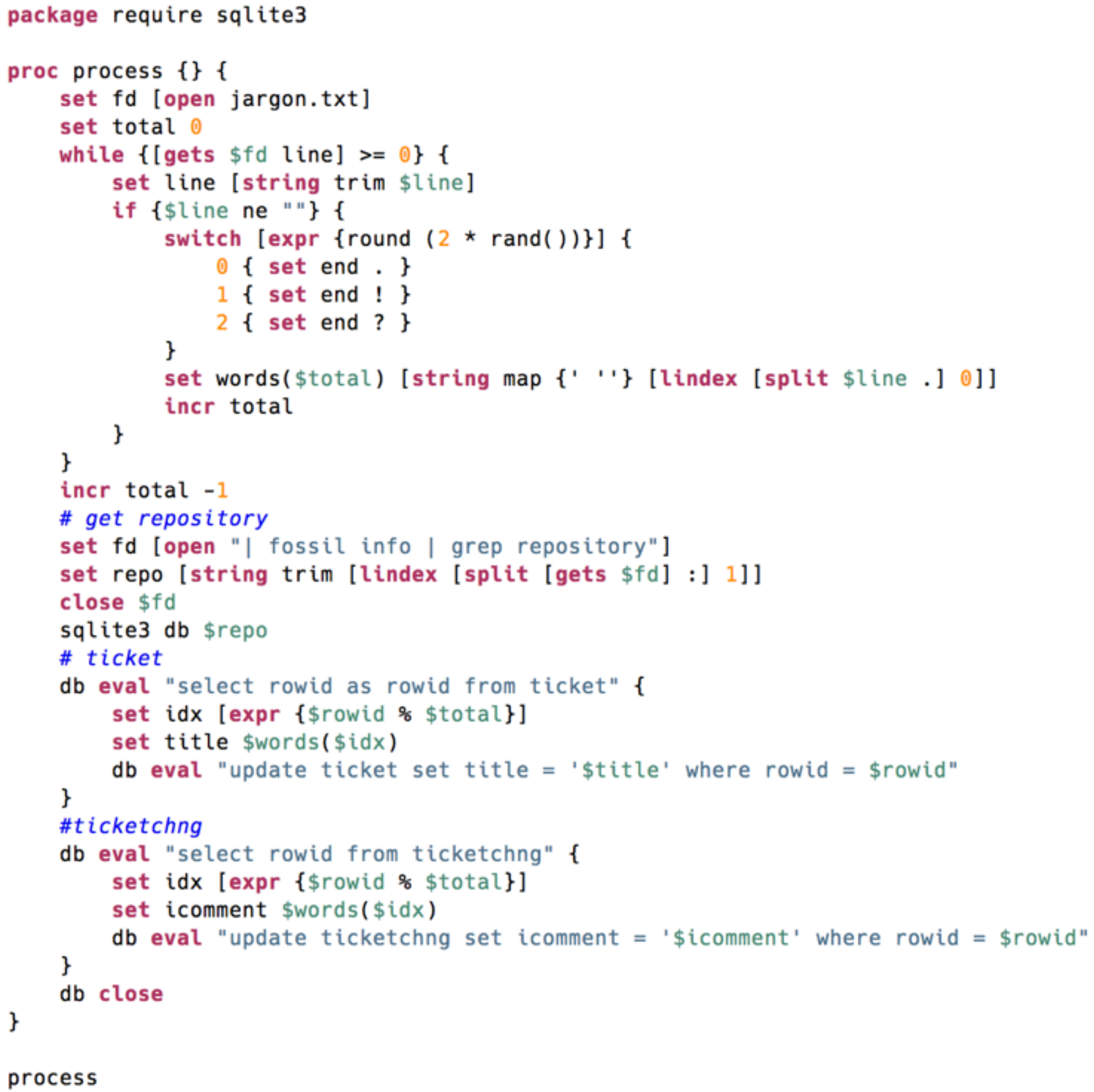
- fossil replace ?
- self-contained
  - script in repo
  - smaller js library
  - js library in repo

Where to now?

- fossil replace ?
- self-contained
- UI improvements
  - unscheduled / completed
  - drag / drop
  - style new/edit screens

Where to now?

- fossil replace ?
- self-contained
- UI improvements
- feedback - is it useful?



```
package require sqlite3

proc process {} {
    set fd [open jargon.txt]
    set total 0
    while {[gets $fd line] >= 0} {
        set line [string trim $line]
        if {$line ne ""} {
            switch [expr {round (2 * rand())}] {
                0 { set end . }
                1 { set end ! }
                2 { set end ? }
            }
            set words($total) [string map {' ''} [lindex [split $line .] 0]]
            incr total
        }
    }
    incr total -1
    # get repository
    set fd [open "| fossil info | grep repository"]
    set repo [string trim [lindex [split [gets $fd] :] 1]]
    close $fd
    sqlite3 db $repo
    # ticket
    db eval "select rowid as rowid from ticket" {
        set idx [expr {$rowid % $total}]
        set title $words($idx)
        db eval "update ticket set title = '$title' where rowid = $rowid"
    }
    #ticketchng
    db eval "select rowid from ticketchng" {
        set idx [expr {$rowid % $total}]
        set icomment $words($idx)
        db eval "update ticketchng set icomment = '$icomment' where rowid = $rowid"
    }
    db close
}

process
```