TODO: Finding unfinished code in flight software FIXME!

RISK. ***What if...?*** *We launch a remote, robotic mission and* ***encounter*** *anomalies due to* ***undiscovered, unfinished code****?*

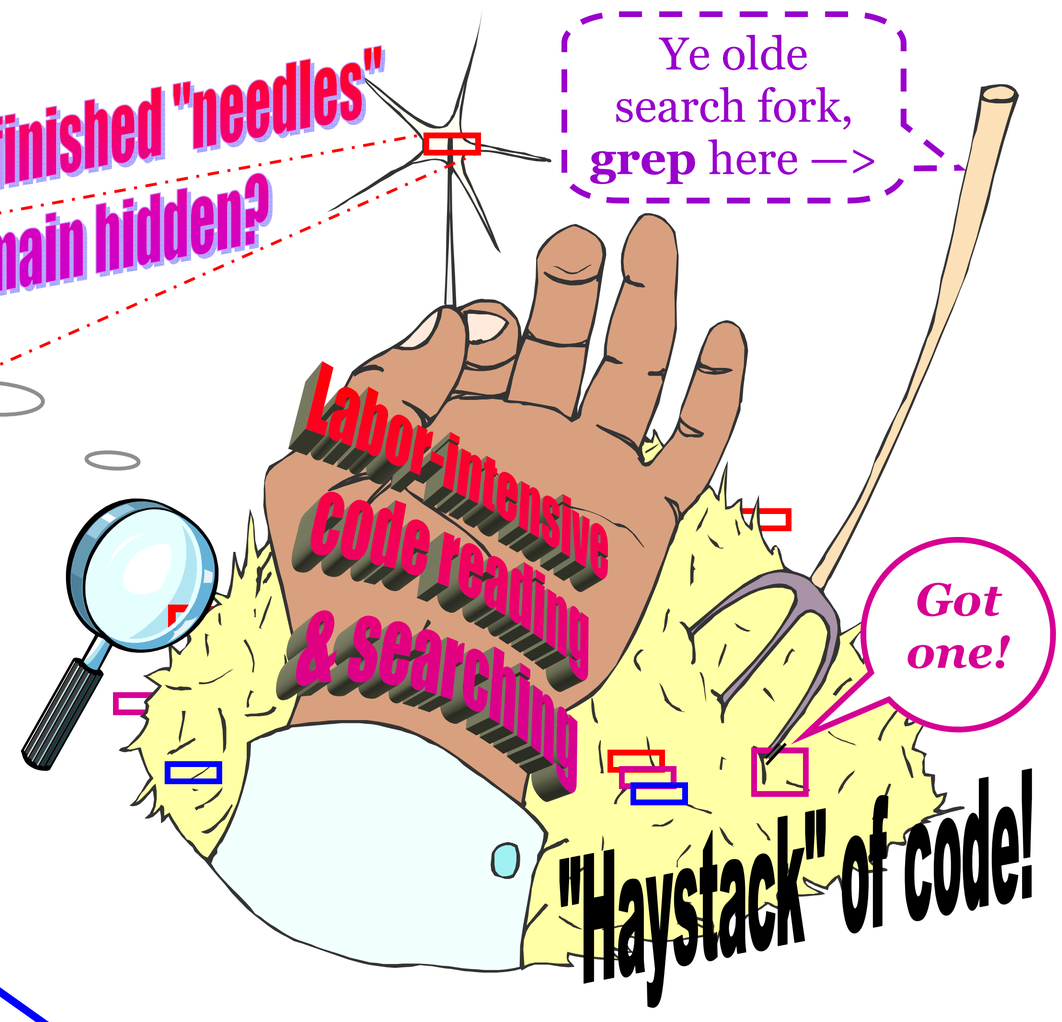
AVOIDANCE. ***What if...?*** *We could* ***implement*** *an effective, reliable method to* ***guard and track unfinished code****?*

TBD

Question: What could be *difficult* about *finding* unfinished code?

Unfinished "needles" remain hidden?

Ye olde search fork, **grep** here —>



A Taxonomy

- Unfinished code
  - Unintentional — Dead ends!
  - Intentional
    - Unmarked — Dead ends!
    - Marked
      - Irregular pattern
      - Regular pattern
        - Passive (comment)
        - Active (macro)

```
/* Example: C comment block */
// Example: C++ comment to EOL
```

```
/* Not yet implemented */
// Partial implementation
/* Change after testing
   with real hardware */
// not certain if correct
//temporary kludge
//kluged for now
// Any other cases to code?
// Is this one outdated?
/* Fred, please fix this */
//approximate, revise later
//skeleton return dummy val
```

spellings may vary

```
/* TODO: R2 */
/* FIXME! */
//HACK HACK!
// TBD
/*TBS*/
/*XXX*/
//unfinished
// STUB
```

```
/* MACRO LIBRARY... */
#define UNFINISHED(notes) {}
/* Many "active" options:
 * Compile-time REPORTS
 * Run-time TRAPS & LOGS
 * Distribution analysis
 */
```

```
UNFINISHED("TODO: FIXME!")
UNFINISHED("Temporary HACK!")
UNFINISHED("Not implemented")
UNFINISHED("Change after test"
   "ing with real hardware")
UNFINISHED("Fred: please fix")
UNFINISHED("skeleton, ret -1")
```

Compiler
Analyzer

UNFINISHED Findings Reports

High effort! x High risk!

Moderate effort x Moderate risk

Low effort x Low risk

Review auto-generated reports

Get busy... reading code... lots and lots of code!

Start searching... and search again... a plethora of patterns!

Answer: *Ease/difficulty* of *finding* unfinished code *depends on...*

High effort

High risk

RISK

AVOIDANCE

UNFINISHED("2011.09.08.0409.KCR")

Lesson: *Placing* **Active** marks in *all* unfinished code to *reduce risk*

Ken Ritchie & **Jake Cox** — Kenneth.C.Ritchie@ivv.NASA.gov, Jacob.T.Cox@ivv.NASA.gov — KEYLOGIC, TASC

NASA POC: **Lisa Downs** Sadie.E.Downs@NASA.gov

NASA Independent Verification and Validation Facility, Fairmont, West Virginia