‘sct-next’

SCT for modern platforms

Presented by Olivier Martin

presented by

ARM®

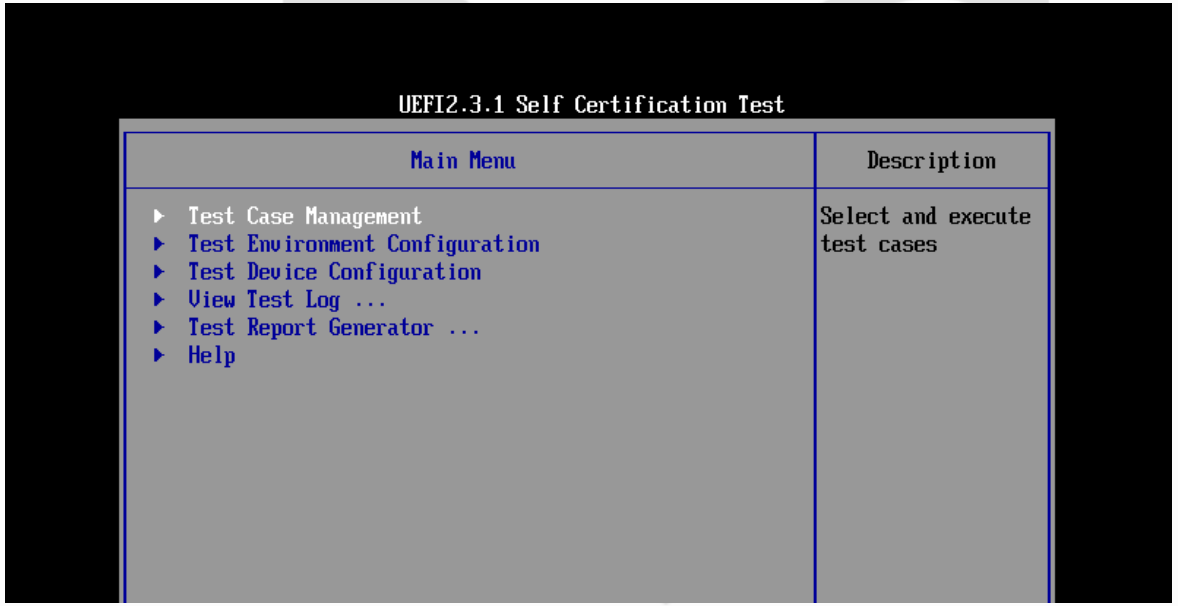
UEFI Plugfest – May 2014

Agenda

- UEFI SCT overview
- Why ‘sct-next’?
- Steps toward ‘sct-next’
- After ‘sct-next’?

UEFI SCT Overview

- **S**elf **C**ertification **T**est
- Driven by **U**EFI **T**est **W**orking **G**roup – *utwg@uefi.org*



What is SCT?

- A Test Framework running in UEFI

  … a set of UEFI drivers and applications

- Test if the UEFI firmware complies with the latest UEFI specification

Why ‘sct-next’?

SCT legacy

- Require EDK Shell (sometimes named as EFI Shell 1.0)

"I think EDK shell is not actively maintained now. We should migrate to UEFI shell ASAP." (answer from one of the owner of EDK Shell)

- Require *EdkCompatibilityPkg*

- Accumulation of years of development

Steps toward ‘sct-next’

New Development Process

- From an obscure ZIP file to an ‘open’ Github repository
  - Allow code review, issue tracker
  - Development more transparent to encourage contributions

- How to contribute:
https://sourceforge.net/apps/mediawiki/tianocore/index.php?title=ArmPkg/HowToContributeSct

Step 1: Clean the code

- Remove dead code & leftovers

```
327  --*/
328  {
329    EFI_STATUS  status;
330    DEBUG ((EFI_D_INFO | EFI_D_LOAD, "wwwwwwwwwwwwwwwwwwww"));
331    EfiInitializeTestLib (ImageHandle, SystemTable);
332    InitializeLib (ImageHandle, SystemTable);
```

```
795
796    if(EFI_ERROR(Status))
797    {
798      EntsPrint(L"Fuck\n");
799          EFI_DEADLOOP();
800    }
```

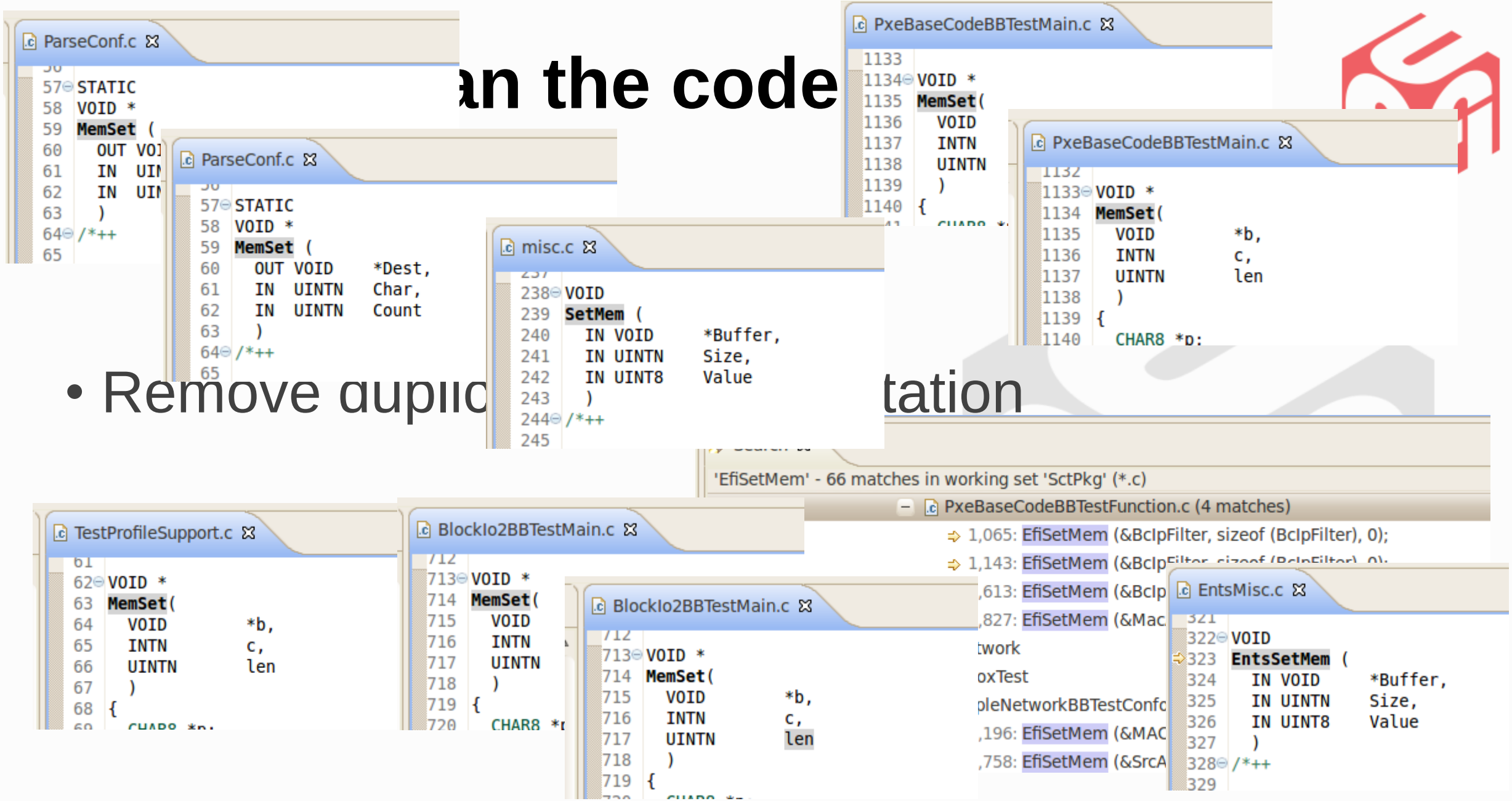
Step 1: Clean the code

- Remove dead code

- Remove duplicated implementation

an the code

- Remove duplic tation



Step 1: Clean the code

- Remove dead code
- Remove duplicated implementation
- New helper library

EfiTestUtilityLib.h

```
491
492 INTN
493 CompareGuid(
494   IN EFI_GUID                       *Guid1
495   IN EFI_GUID                       *Guid2
496   );
497
```

Debug.c

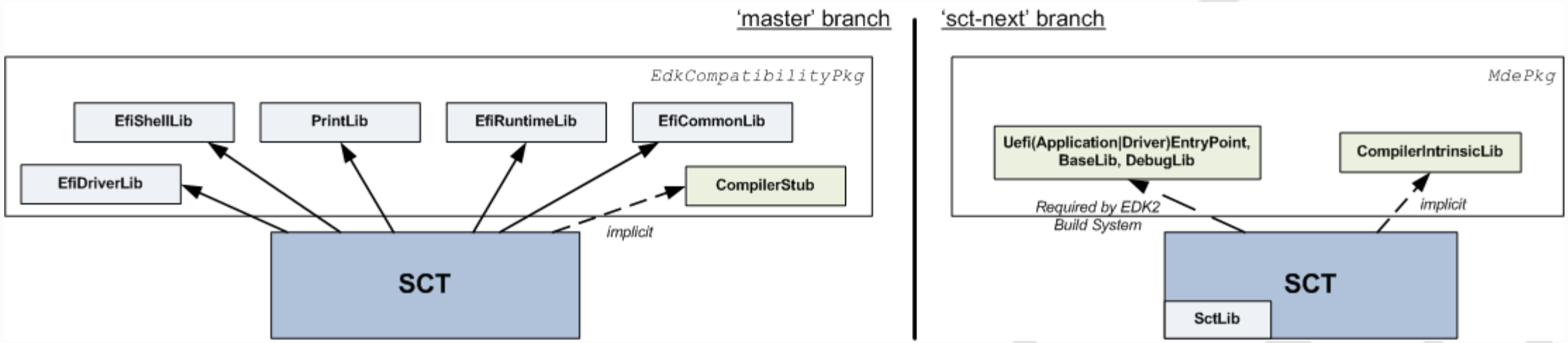
```
60
61 BOOLEAN
62 EfiCompareGuid (
63   IN EFI_GUID *Guid1,
64   IN EFI_GUID *Guid2
65   )
66 {
```

Step 2: Run SCT on EFI Shell 2.0

- Abstraction to still be able to run on EDK Shell

Step 3: Disconnect from EdkCompatibilityPkg

- Reduce dependency on the legacy framework



- Avoid to maintain legacy code

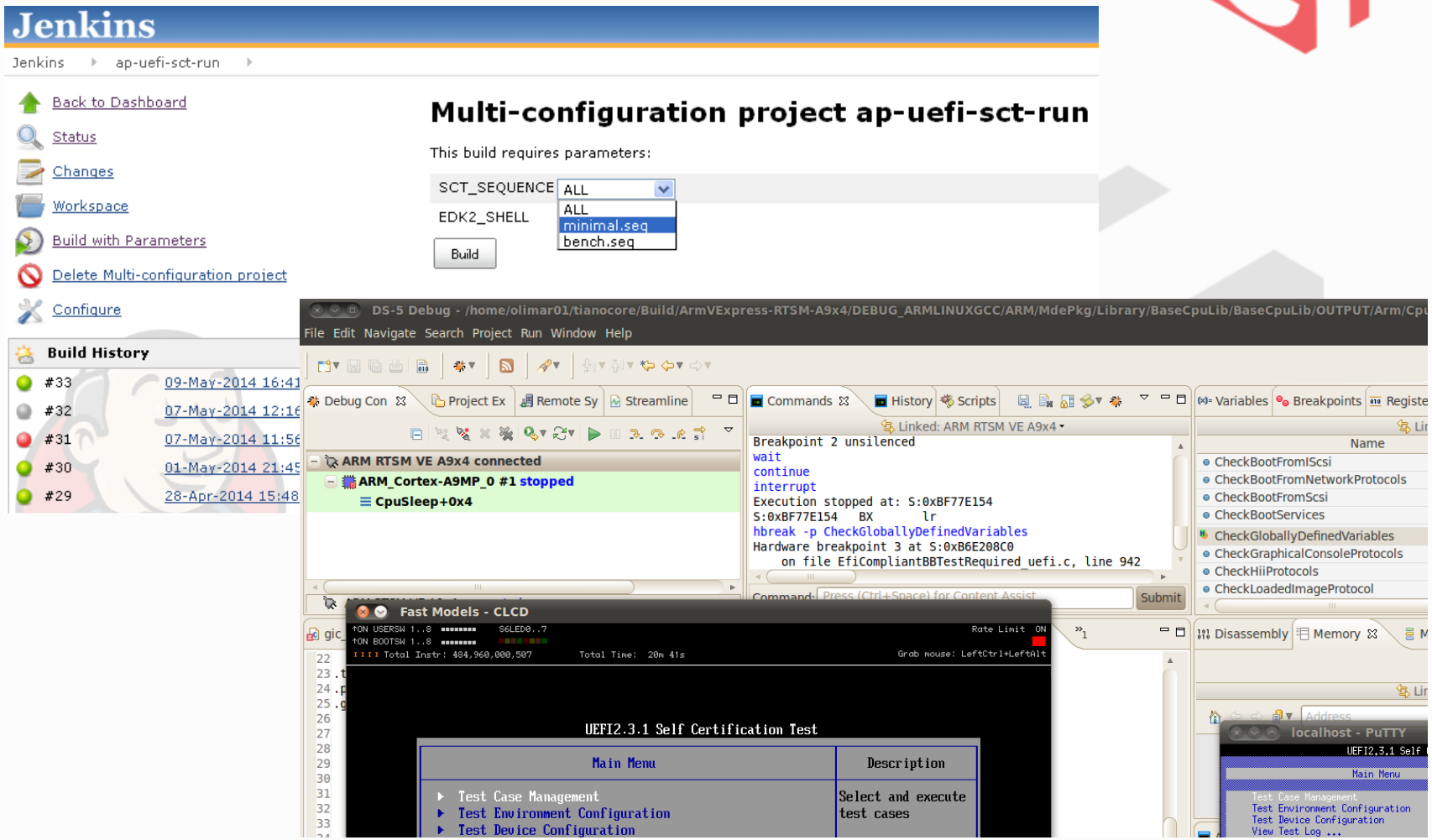
Step 4: Performance improvement

- Run SCT faster to run SCT more often

- To not use the excuse of the long SCT execution time to not contribute to SCT…

Step 5: Validation

- Automation
- Regression
- Debugging



Results [1/2]

- Parity between the ‘master’ and ‘sct-next’ branches
- Found & Fix 12 defects in EFI Shell 2.0
- Found limitation in UEFI Shell 2.0 spec
- Tried (and documented) new Github process
- Inconsistencies in UEFI SCT with the UEFI spec

Results [2/2]

- **`$ git diff master..HEAD --shortstat`**

  `1745 files changed, 66752 insertions(+), 96577 deletions(-)`

- Performance improvement … work-in-progress

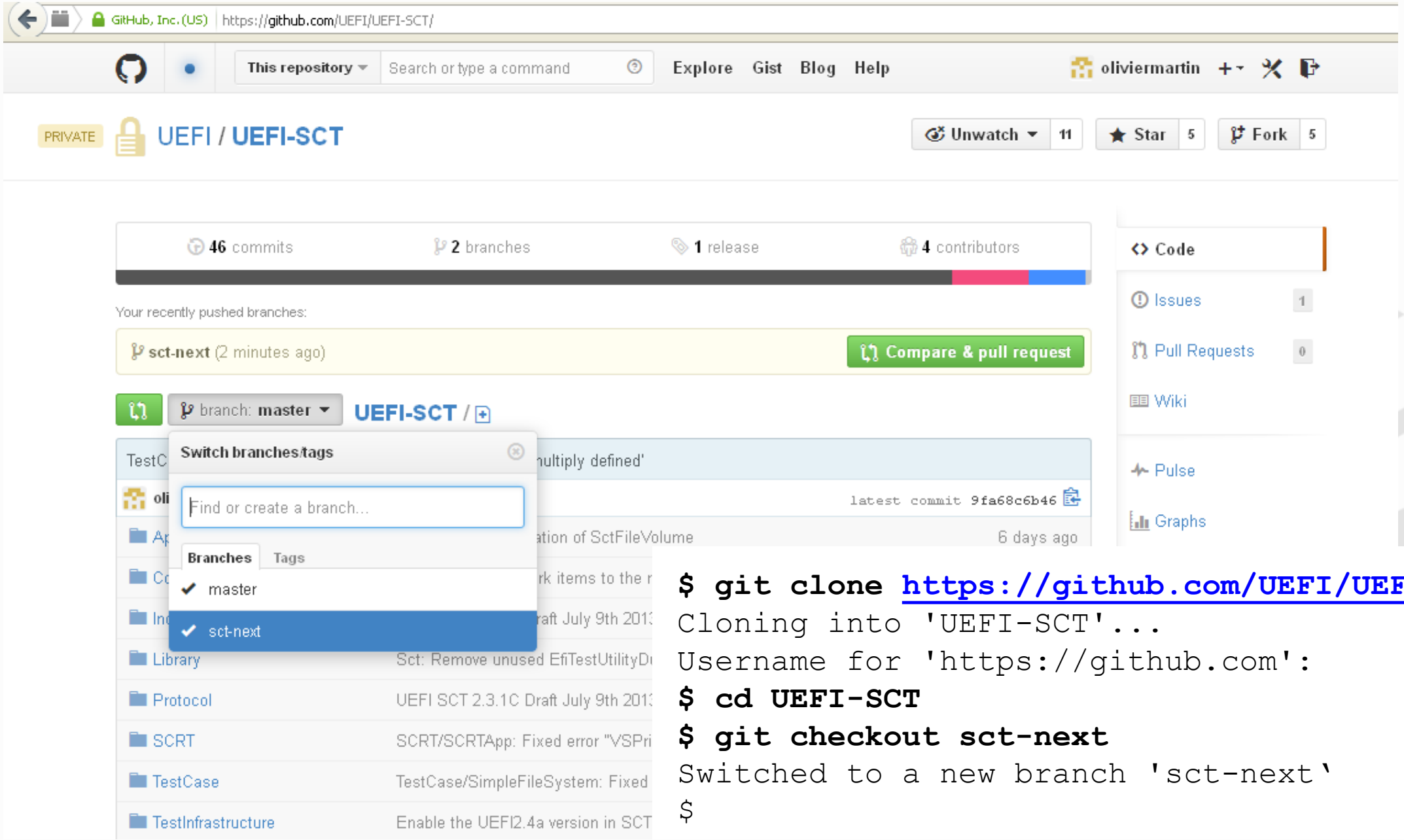
After ‘sct-next’?

Next official SCT release?

- Need contribution from UEFI members

… what is a contribution?

  - Code changes
  - Code review
  - Documentation
  - Test




```
$ git clone https://github.com/UEFI/UEFI-SCT
Cloning into 'UEFI-SCT'...
Username for 'https://github.com':
$ cd UEFI-SCT
$ git checkout sct-next
Switched to a new branch 'sct-next'
$
```

Future of ‘sct-next’?

- Do we really need to run SCT from a **Shell**?
- Remove **ALL** the dependencies around *SctPkg*
- More than a **UEFI** spec conformance test…
  - **EFI Shell** spec?
  - **ACPI** spec?
  - Vendor specific test (eg: (**ARM**) **S**erver **B**ase **S**pecification **A**rchitecture) ?

Questions ?!?

Resources:

- How to contribute to SCT?
http://sourceforge.net/apps/mediawiki/tianocore/index.php?title=ArmPkg/HowToContributeSct

- How to debug SCT?
http://sourceforge.net/apps/mediawiki/tianocore/index.php?title=ArmPkg/Sct

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by