Developing New Element with GstCheck

Jeongseok Kim <justin.joy.9tail.com>
LG Electronics, Inc.

Gnome Asia Summit 2014

Who is Jeongseok?

- From South Korea
- A Software Engineer at LG Electronics
- A Gnome User, Just beginner for contribution
- A Beijing Tourists

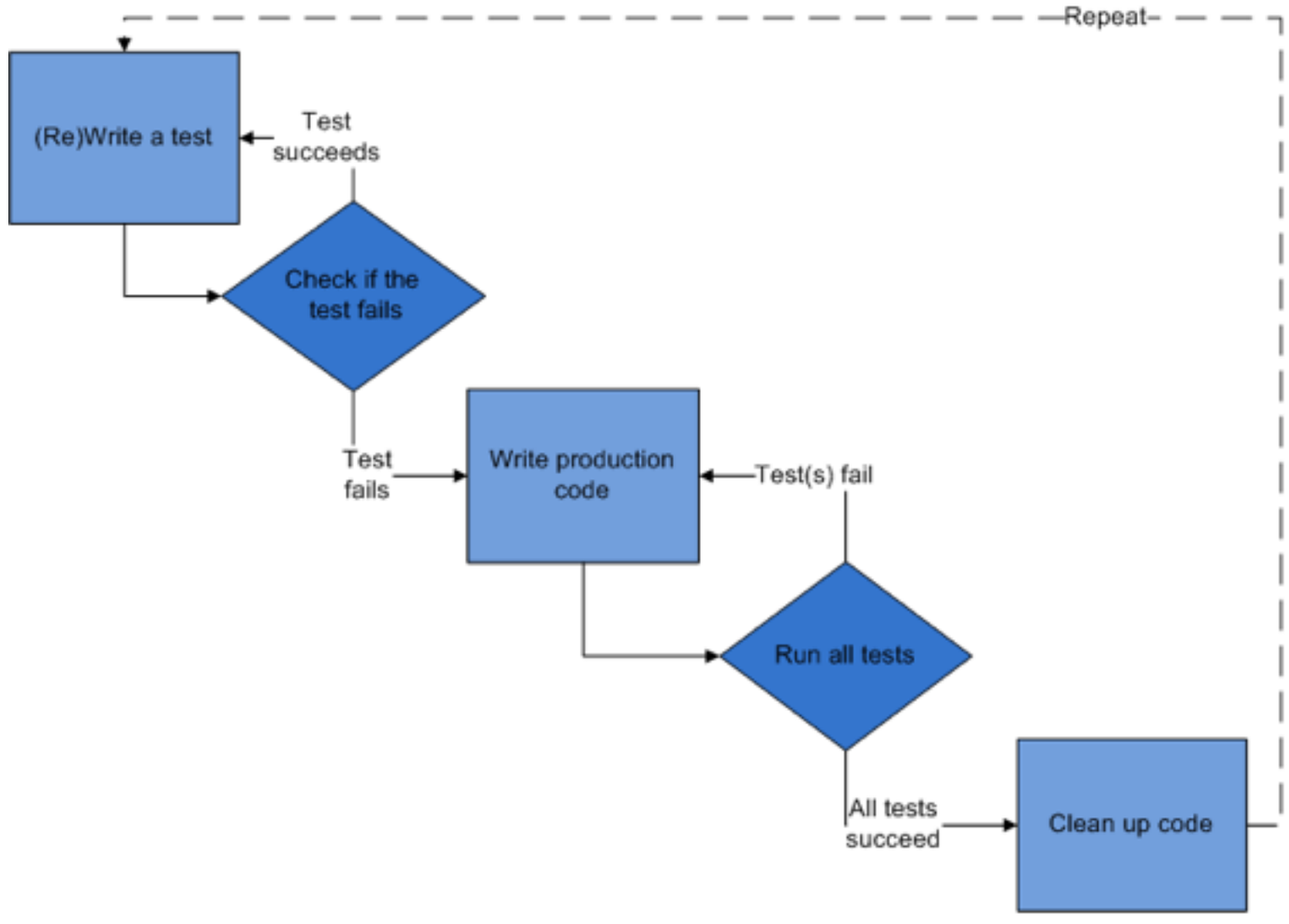
TOC

- Related Works - Background
- What is GstCheck
- DynAppSrc - tiny, but highly required new element
- Interaction between implementation and testing
- About final structure
- QnA

Related Works -Background

- Software Development Process
  - TDD (Test-Driven Development)
    - The more test, The better code
- BDD (Behaviour-Driven Development)
  - TDD with domain specific design concepts

Test Driven Development



<WikiPedia - Test Driven Development>

Behavior Driven Development

(with example)

Describe in a plain text	**Test Code**
• I need a calculator.	calculator = new Calculator
• If I enter 2 and 3, it will show 5.	result = calculator.enter (2, 3) assert (result == 5)

More Details

- http://cukes.info/

GstCheck from libcheck

- Actually, GstCheck is not a feature only for gstreamer.
- GstCheck is a wrapper of libcheck which is one of unit test frameworks.

```
#include <check.h>
#include "../src/money.h"

START_TEST (test_money_create)
{
  Money *m;
  m = money_create (5, "USD");
  ck_assert_int_eq (money_amount (m), 5);
  ck_assert_str_eq (money_currency (m), "USD");
  money_free (m);
}
END_TEST
```

<example from libcheck tutorial>

Comparison between libcheck and GstCheck

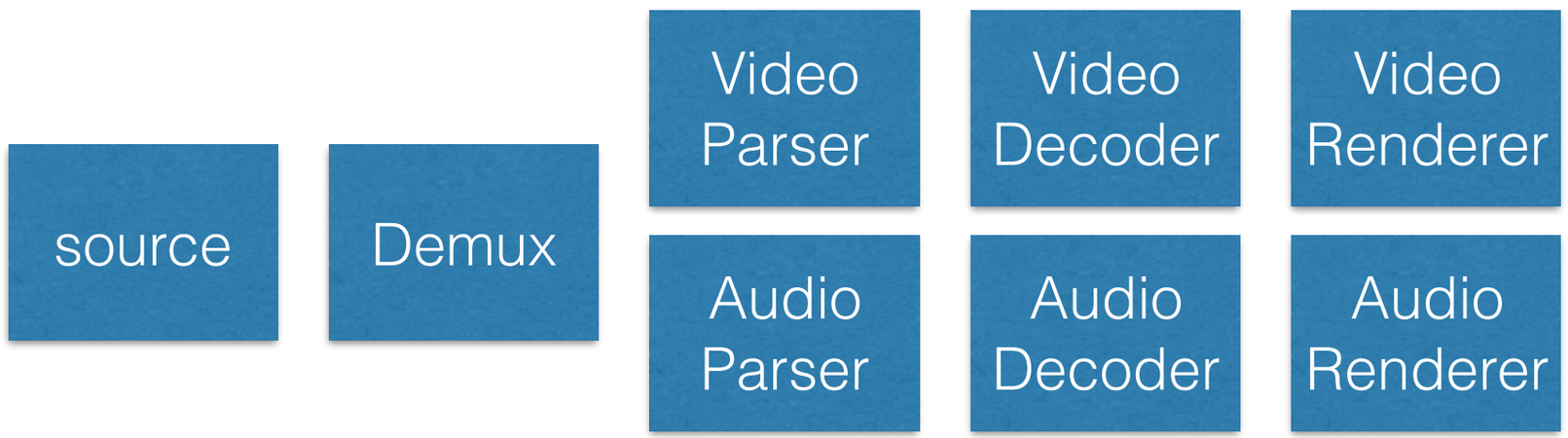
```
#include <check.h>
#include "../src/money.h"

START_TEST (test_money_create)
{
  Money *m;
  m = money_create (5, "USD");
  ck_assert_int_eq (money_amount (m), 5);
  ck_assert_str_eq (money_currency (m), "USD");
  money_free (m);
}
END_TEST

Suite *money_suite (void)
{
  Suite *s = suite_create ("Money");

  /* Core test case */
  TCase *tc_core = tcase_create ("Core");
  tcase_add_test (tc_core, test_money_create);
  suite_add_tcase (s, tc_core);

  return s;
}
```

```
#include <gst.h>

GST_START_TEST (test_money_create)
{
  Money *m;
  m = money_create (5, "USD");
  fail_unless (money_amount (m) == 5);
  fail_unless_equals_string (money_currency (m), "USD");
  money_free (m);
}
GST_END_TEST

static Suite *
money_create_suite (void)
{
  Suite *s = suite_create (“money_create");
  TCase *tc_chain;

  tc_chain = tcase_create ("general");
  tcase_add_test (tc_chain, test_money_create);
  suite_add_tcase (s, tc_chain);

  return s;
}
```

DynAppSrc Element

Back to the gstreamer

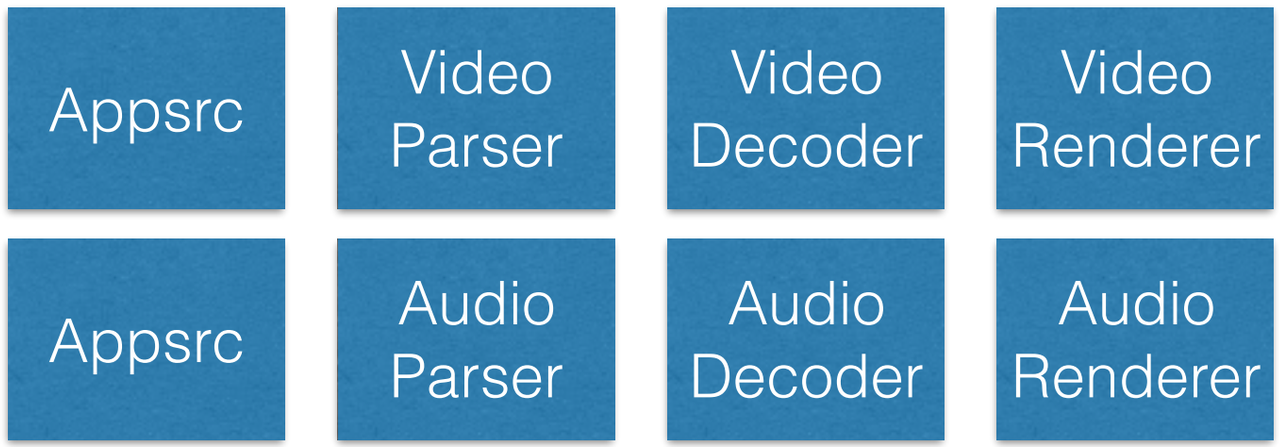
- A Kind of Enhancement feature for playbin
  - https://bugzilla.gnome.org/show_bug.cgi?id=725187
- Normally, a pipeline handle data from an octet-stream.



DynAppSrc Element

(requirements)

- Each element stream is given for various reasons.
  - DRM (decrypted by outside components)
  - Strictly closed data feeding library
  - Or, Legacy?



DynAppSrc Element

- Handling multiple appsrcs by a source element
- Including features to
  - Create AppSrc instance by signal action
  - Deliver SEEK event properly
  - Be compatible with playbin

DynAppSrc Test Cases

1. How to create DynAppsrc
2. How to handle internal Appsrc elements
3. Validate internal Appsrc instances
4. Add constraint user actions
5. Seek event
6. EOS event

Let’s see the real code in Git

https://github.com/justinjoy/gst-plugins-lp

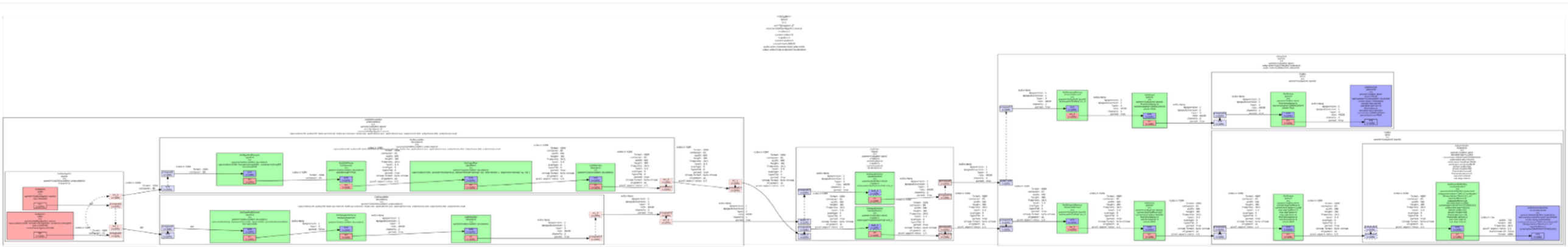
```
a8a5bea tests: dynappsrc: Update test case for EOS event
48f4f53 tests: dynappsrc: Add test case for EOS event
c659596 tests: dynappsrc: Add test case for seek-event
6e92632 tests: dynappsrc: Add test case for upstream event
55a4135 tests: dynappsrc: Add test case for repeatedly setting appsrc
4fbea1b tests: dynappsrc: prevent to add another appsrc in paused state
00bd2b5 tests: dynappsrc: change new-appsrc arguments
f8c5fe6 tests: dynappsrc: explain more about test case
2a49015 tests: dynappsrc: prevent infinite loop during iterating srcpad
d5d7d06 tests: dynappsrc: add test for verifying dynamic element
48b3386 dynappsrc: add test suite to check initial structure
```

```
e93d76a dynappsrc: EOS event needs to be spread into each source
ab7facb dynappsrc: seek event needs to be spread into each source
60d7fd3 dynappsrc: Add dynappsrc example documents
2a40fbb dynappsrc: remove memory leakage
c3141af dynappsrc: initialize n-source property when state is changed
fbbd6b9 dynappsrc: Fix memory leakage of srcpad
dd20f85 dynappsrc: Add event and query handler fuctions
bddad90 dynappsrc: use no_more_pad function
b4c8f6d dynappsrc: Add n-source property
50aadb6 dynappsrc: Don't leak template in setup_source
40d6326 dynappsrc: Add source configuration
cd9110e docs: Add dynappsrc documentation
dd366a6 dynappsrc: Add new-appsrc signal action
9d4821f dynappsrc: Add dynamic appsrc element
```

Conclusion

- Pros
  - Code quality
  - Memory Leakage Detection
  - Easy to define the next step
- Cons
  - Understanding & sharing process
  - Time challenge

Final Structure



“Questions and Answers”

Or Let me know Beijing Attractions

Thank you

References

- http://en.wikipedia.org/wiki/Test-driven_development
- http://en.wikipedia.org/wiki/Behavior-driven_development
- http://cukes.info/
- http://check.sourceforge.net/