



TU Dresden – Media Center

Visual Regression Testing

Paul Grunewald

Oct 9, 2018



TU Dresden and Plone

Some facts on TU Dresden

- one of the biggest universities in eastern Germany
- 35,000 students and 6,600 employees
- about 3,500 editors

Our Plone deployment

- Plone 4 (Archetypes)
- Large scale: 600k content objects, 2.5 mio page views/month
- Heavily customized Plone UI (for visitors/editors)
- High emphasis placed on compliance with the Corporate Design
- Testing, testing, testing... over 400 Robot tests!

Side note: For an excellent in-depth technical report of our deployment, see last year's presentation in Barcelona. :)





	TU DRESDEN	STUDIES	RESEARCH	CAREER	COOPERATION	
LAG DER LEHRE STUDIES / COURSES 5, 12, 2018 JUDRESDER JU	H			KCELLEN MORE RESDEN J	CE REGION	Landon and a second and a

Ø

Important Information

Faculties & Units



The "Service Center Studies" offers answer

your application to the TU Dresden and will

point you towards the right contact person.

Frontdesk are open from Monday to Friday

You can get in touch by telephone +49 351

servicecenter studium@tu-dresden.de.or

visit us in person in the SLUB foyer.

463 42 000 or by email

>Read more

to all questions regarding your studies.

The TU Dresden student hotline and the



APPLY FOR A RELOCATION

The city of Dresden pays a relocation

allowance to students, who move their

primary place of residence to Dresden

from another German city. The relocation

to the primary place of residence must be

•0

completed within the period of one year

ALLOWANCE AT THE

STUDENTS' UNION

(January and December).

Read more



INTERNATIONAL TUTOR PROGRAMME

Would you like to spend one or two semester or even your entire studies at TU Dresden and require assistance in organisational matter regarding the start of your studies? The tutors of the International Office are eager to help!

>Read more

Language 🍘 Search 🔎 Internal 🥑

Regression Testing What's that?

Regression testing (rarely non-regression testing) is re-running functional and nonfunctional tests to ensure that previously developed and tested software still performs after a change. If not, that would be called a **regression**.

Source: Wikipedia, Nov 2019

Some reasons that lead to regression

- Changing things your don't quite understand
- Insufficient documentation
- Convoluted dependencies
- Changed bit is complicated (e.g. nested logic)
- ... and are lot more!





Website before



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur elementum maximus urna, ac ultrices tellus viverra vel. Ut venenatis, lectus a rutrum varius, diam augue aliquet mi, ac tempus ante sapien porta risus. Nam neque mi, efficitur eget quam auctor, suscipit rhoncus est.





Designer incorporating the latest changes...







Screenshot after ... spot the difference? It is the same, right?



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur elementum maximus urna, ac ultrices tellus viverra vel. Ut venenatis, lectus a rutrum varius, diam augue aliquet mi, ac tempus ante sapien porta risus. Nam neque mi, efficitur eget quam auctor, suscipit rhoncus est.





WRONG!



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur elementum maximus urna, ac ultrices tellus viverra vel. Ut venenatis, lectus a rutrum varius, diam augue aliquet mi, ac tempus ante sapien porta risus: Nam neque mi, efficitur eget quam auctor, suscipit rhoncus est.





WRONG!



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur elementum maximus urna, ac ultrices tellus viverra vel. Ut venenatis, lectus a rutrum varius, diam augue aliquet mi, ac tempus ante sapien porta risus. Nam neque mi, efficitur eget quam auctor, suscipit rhoncus est.





Visual Regression Testing Typical workflow for writing tests







Visual Regression Testing Several tools and cloud services

There exists a multitude of solutions, that provide visual regression testing.



BUT many solutions do not fit in our testing environment or fail due to other reasons (unmaintained, too old, too expensive)





Visual Regression Testing Our testing environment

- Jenkins CI system running our tests every night
- Robot Framework (via plone.app.robotframework)
- robotframework-selenium2screenshots
- Robot Framework Integration Plugin
 - read reports directly in Jenkins
- Testing against Firefox ESR versions
- Building test sites for manual checking:
 - develop \rightarrow every day
 - release \rightarrow whenever an update is scheduled

All	Ansible	MOVING	Plone Portale	WebCMS-Features	WebCMS2-Develop	WebCMS2-Master	WebCMS2-Release			
s	w	Name 1			Last Success		Last Failure	Last Duration	Robot Results	
•	*	webcms2-sphinx-develop				16 hr - <u>#602</u>		7 mo 2 days - <u>#390</u>	13 min	
•		webcms2-tud.addons.admin-develop				1 day 15 hr - <u>#1</u>	103	15 hr - <u>#1104</u>	32 min	23 / 24 passed 🛔
•	4	webcms2-tud.addons.chat-develop				16 hr - <u>#445</u>		3 days 16 hr - <u>#442</u>	14 min	8 / 8 passed 🛔
	4	webcms2-	tud.addons.ckedit	orplugins-develop		17 hr - <u>#1074</u>		2 days 17 hr - <u>#1072</u>	18 min	21 / 21 passed 🛔
	-	webcms2-	tud.addons.datag	ridfield-develop		1 day 16 hr - <u>#9</u>	69	16 hr - <u>#970</u>	12 min	0 / 1 passed 🛔
•	*	webcms2-	tud.addons.protec	ction-develop		15 hr - <u>#1061</u>		1 mo 8 days - <u>#1023</u>	14 min	6 / 6 passed 🛔
•	*	webcms2-	tud.addons.redire	ct-develop		16 hr - <u>#973</u>		13 days - <u>#960</u>	14 min	8 / 8 passed 🛔
•	*	webcms2-	tud.boxes.base-de	evelop		15 hr - <u>#1063</u>		1 mo 28 days - <u>#1005</u>	12 min	1 / 1 passed 🛔
		webcms2-	tud.boxes.webcm	s-develop		12 days - <u>#826</u>		16 hr - <u>#1078</u>	4 hr 53 min	228 / 233 passed 🛔
	*	webcms2-	tud.content.webcr	ms-develop		17 hr - <u>#1093</u>		8 days 17 hr - <u>#1085</u>	22 min	37 / 37 passed 🛔
	*	webcms2-	tud.profiles.webcr	ns-develop		17 hr - <u>#989</u>		26 days - <u>#963</u>	23 min	14 / 14 passed 🛔
•	*	webcms2-	tud.theme.webcm	s2-develop		16 hr - <u>#1096</u>		10 days - <u>#1086</u>	31 min	39 / 39 passed 🛔
•	*	webcms2-	wildcard.foldercor	ntents-develop		17 hr - <u>#1083</u>		6 days 17 hr - <u>#1077</u>	16 min	13 / 13 passed 🛔





Visual Regression Testing New Robot Framework keyword

Test vs Reference

- \${selector} ... CSS selector of to be tested element
- \${name} ... name of tested element (used in the filename of the screenshot)
- \${ref_path} ... location where to save/load screenshots
- \${update} ... True? => Accept screenshot as reference, always PASS

False? => Check current against reference screenshot

Consequences in workflow:

Whenever some change is agreed as new expected output, set update=True, re-run tests and set update=False.







Visual Regression Testing Our implementation: Detecting changes

Comparision between current and reference screenshot:

- 1. Determine the maximum width & height of both screenshots
- 2. Calculate the percentage of non-identical pixels
- 3. Pass the test, if the percentage is below a certain threshold

Can be easily replaced with any better performing algorithm.





Visual Regression Testing

Demo







Visual Regression Testing Problems & Solutions

- Miniscule differences in font rendering between operating systems and browsers
 - Use the same OS and browser as CI system (e.g. via vagrant)
- Screenshot contains parts that are not controllable (timestamps, remote content, animations)
 - time related issues? use package freezegun in your setup of the testing layer
 - black out those parts
- **Storage** (keep the reference screenshots in the repository?)
 - use submodules with shallow clone to have only the latest version of the screenshots





Visual Regression Testing Next steps

- Better workflow for selecting reference screenshots
- Better algorithms for detecting changes in the screenshots
- Finish & cleanup the package and put it on PyPi





Visual Regression Testing

Thank you どうもありがとうございました





