

11-11-2017

Recoding “Guide on the Side” to collect learner performance data: Collaboration, customization, and assessment

Graham Sherriff
University of Vermont

Follow this and additional works at: <http://scholarworks.uvm.edu/libfacpub>



Part of the [Information Literacy Commons](#)

Recommended Citation

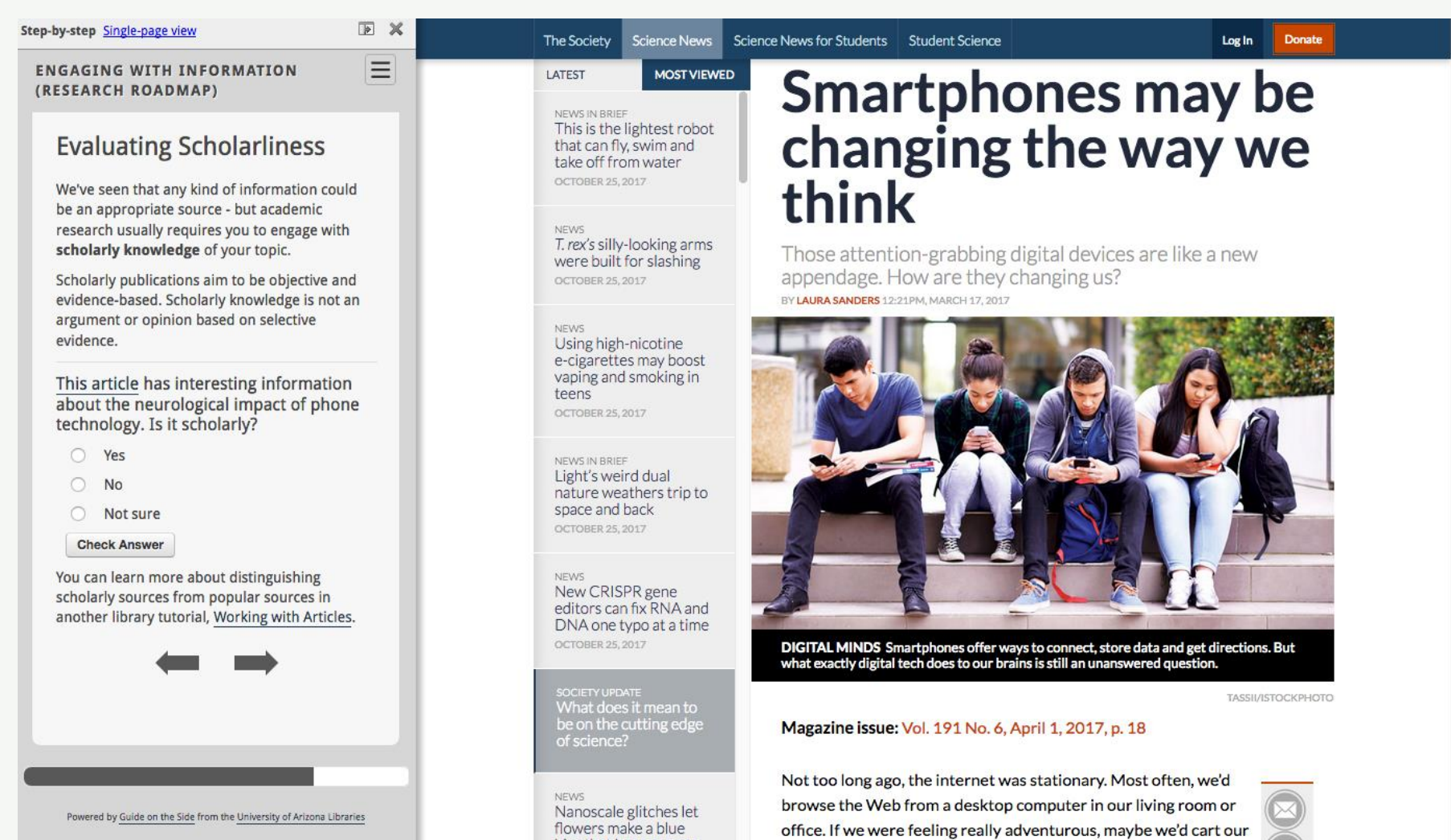
Sherriff, Graham, "Recoding “Guide on the Side” to collect learner performance data: Collaboration, customization, and assessment" (2017). *University Libraries Faculty and Staff Publications*. 56.
<http://scholarworks.uvm.edu/libfacpub/56>

This Poster is brought to you for free and open access by the University Libraries at ScholarWorks @ UVM. It has been accepted for inclusion in University Libraries Faculty and Staff Publications by an authorized administrator of ScholarWorks @ UVM. For more information, please contact donna.omalley@uvm.edu.

Recoding “Guide on the Side” to collect learner performance data

Graham Sherriff, Instructional Design Librarian, University of Vermont (UVM)

Guide on the Side tutorials generate a lot of learner performance data (students’ quiz answers and tutorial scores). How can we analyze it?



Background

Guide on the Side (GotS) is an open source platform for split-screen tutorials using real-time web content, developed by the University of Arizona Libraries.

The University of Vermont (UVM) libraries have used GotS to flip information literacy instruction since 2013. Our GotS tutorials support first-year information literacy instruction and a range of discipline-specific library instruction.

- Interactive learning experiences
- Authentic web content
- User-controlled pacing
- Designed for formative learning
- Option to include summative quiz at the end
- A simple WYSIWYG authoring interface
- No outdated screen captures
- Scalable (~5,000 completions per semester at UVM)
- Minimal technical maintenance
- No purchase or licensing fees

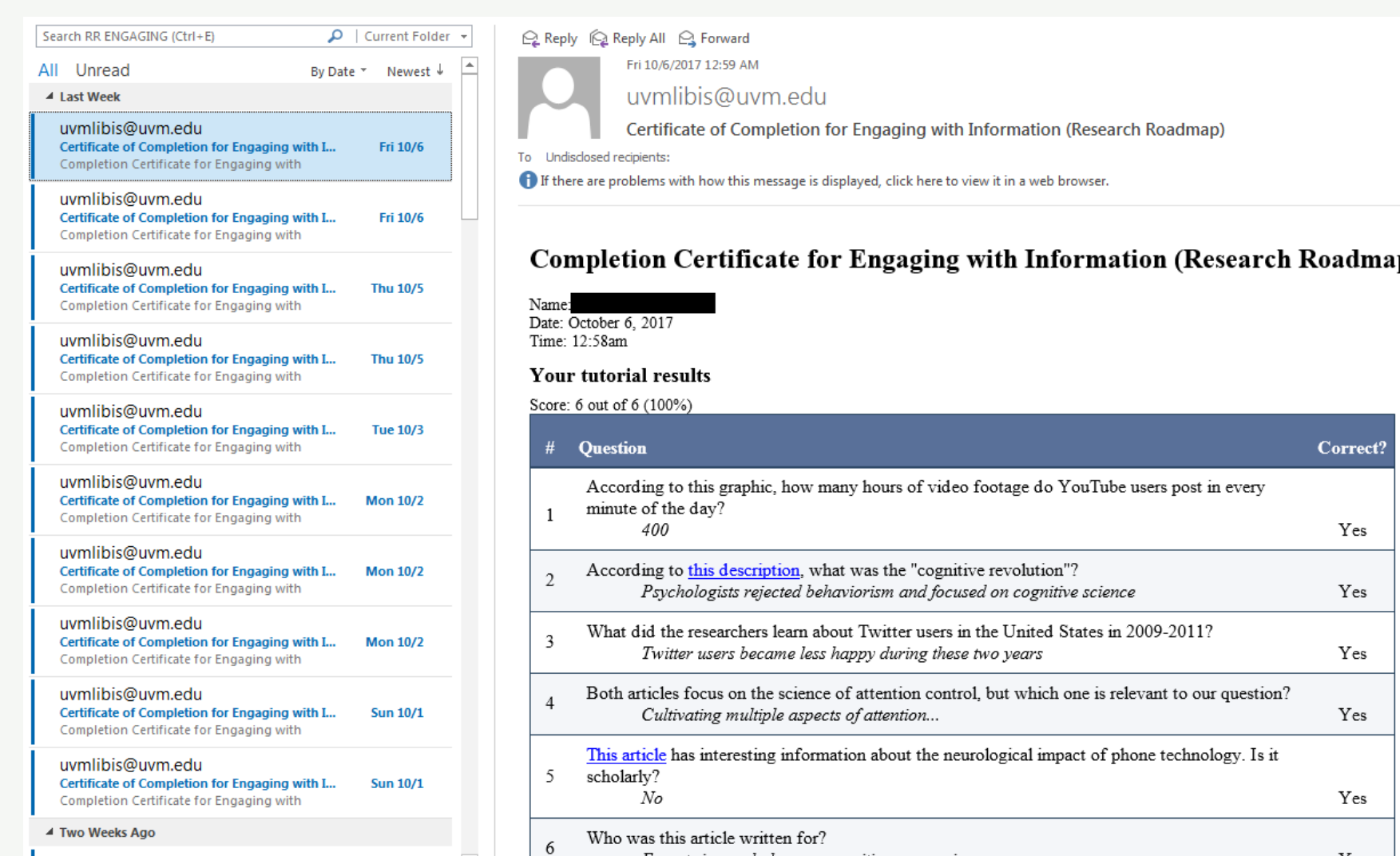
Problem

No features for automated aggregation or reporting of learner performance data (responses to questions and tutorial scores).

Data are captured only as “certificate” emails to the student and the tutorial owner/administrator.

Assessment requires manual data entry, which is time-intensive and susceptible to errors.

The alternative platform is less supportive of formative or exploratory learning, and is license-based.



Specifications

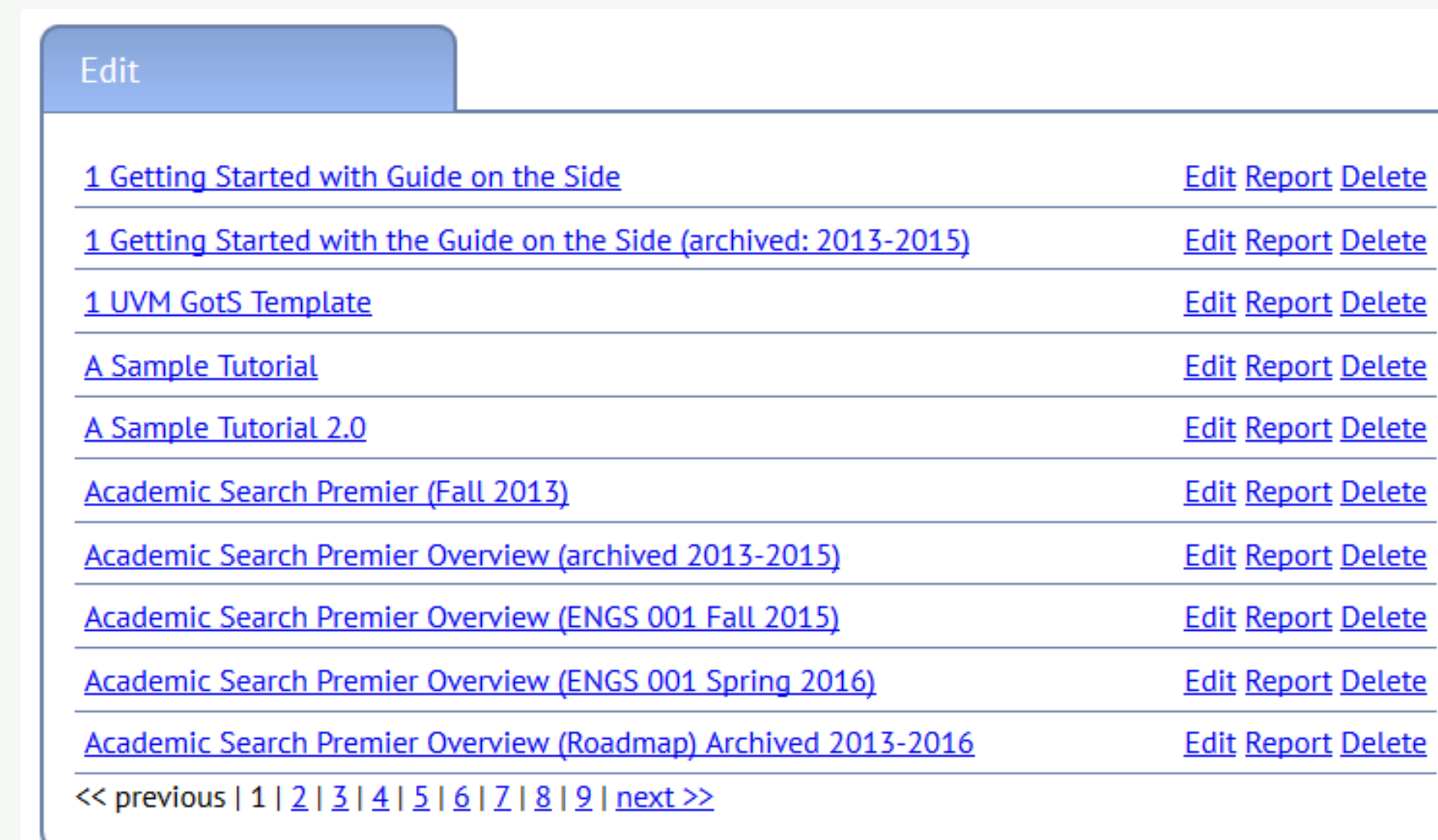
- Automated aggregation of date- and time-stamps, multiple-choice responses, open responses, tutorial scores, and quiz scores
- Anonymized data: no collection of student names, email addresses, or IP addresses
- Reports that allow statistical analysis
- Admin-only access to reports

Collaboration and customization

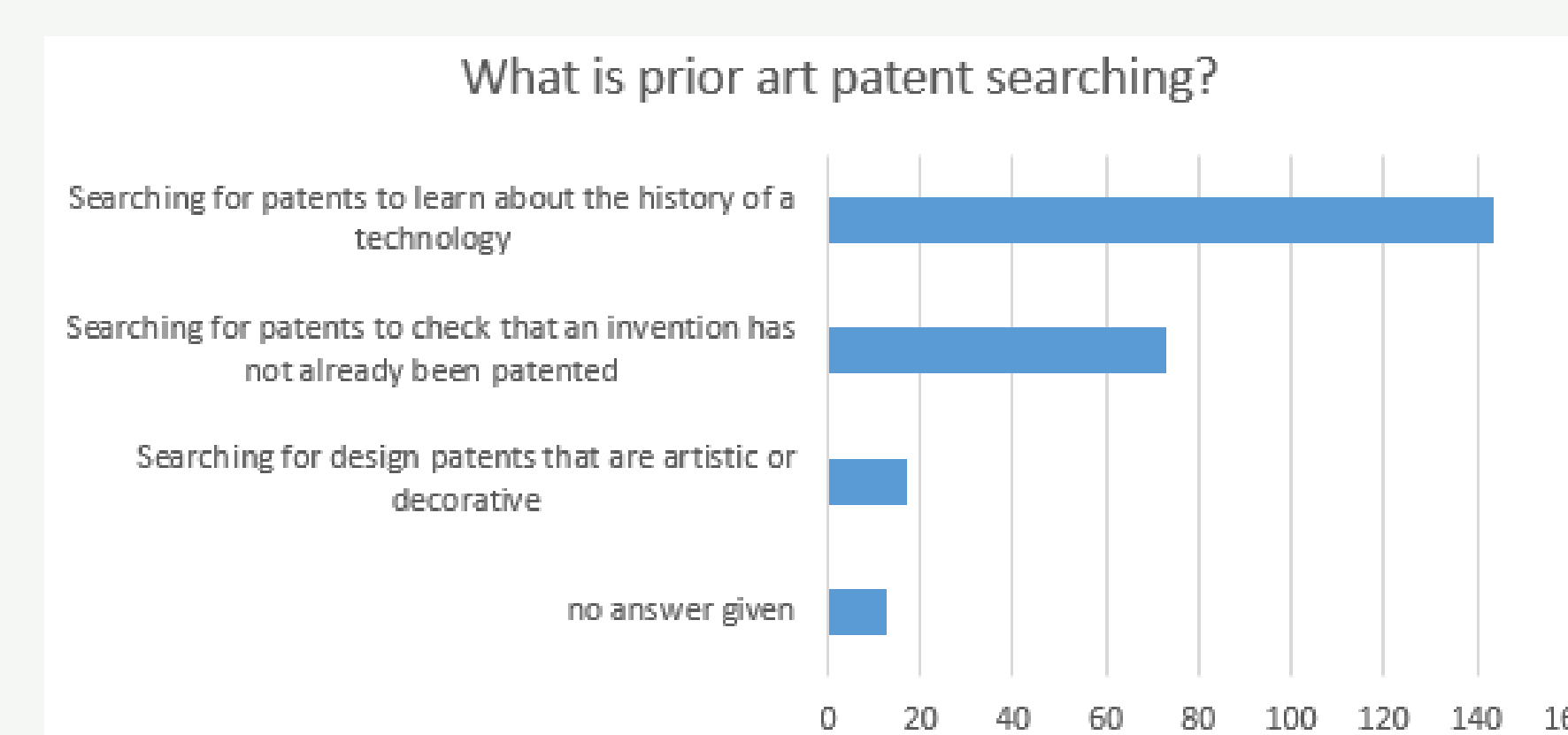
Wesley Wright (UVM Center for Teaching and Learning) added custom code to the PHP files in our local GotS installation and the MySQL files that store the learner performance data.

Our installation now aggregates the specified data points for each completed tutorial.

Reports are accessible to our GotS admins via a simple “Report” link that the code adds to each tutorial in the admin interface.



Data are reported in .csv format, and can be manipulated and analyzed in Excel.



Outcomes: for UVM libraries

Data aggregation and reporting are more accurate and not susceptible to errors created by manual data entry.

Data aggregation and reporting are instant and more time-efficient. Aggregation no longer requires student worker hours.

Data analysis can be done on a larger scale. Analysis is no longer limited to small datasets or sampling. Large-scale analysis can be applied across multiple sections or classes, or longitudinally.

Renewed commitment to GotS as the UVM libraries’ supported platform for online tutorials.

Outcomes: beyond UVM libraries?

Proof of concept: the custom code offers a stable set of features for aggregating and reporting GotS learner performance data.

These enhancements would make this open source program a more viable option for libraries considering a platform for split-screen tutorials.

GotS developers at the University of Arizona are reviewing the code.

Acknowledgments

- Wesley Wright (IT Specialist, UVM Center for Teaching and Learning)
- Daisy Benson (Instruction Coordinator, UVM Bailey/Howe Library)
- Lyman Ross (Systems Librarian, UVM Bailey/Howe Library)
- University of Arizona Libraries