Lessons Learned from Data Defects

Rick Rogers, President
Fenestra Technologies Corp.
(240) 535-2064
rick@fenestra.com

“Mistakes are the portals of discovery” – James Joyce

Outline

- Operational Context
  - Economic Census
  - GIDS Designer
  - GIDS Surveyor
- Data Defect Anecdotes
  - Major defect, needed to be fixed immediately
  - Minor defect, did not need to be fixed
  - Process defect, tricky to find and fix

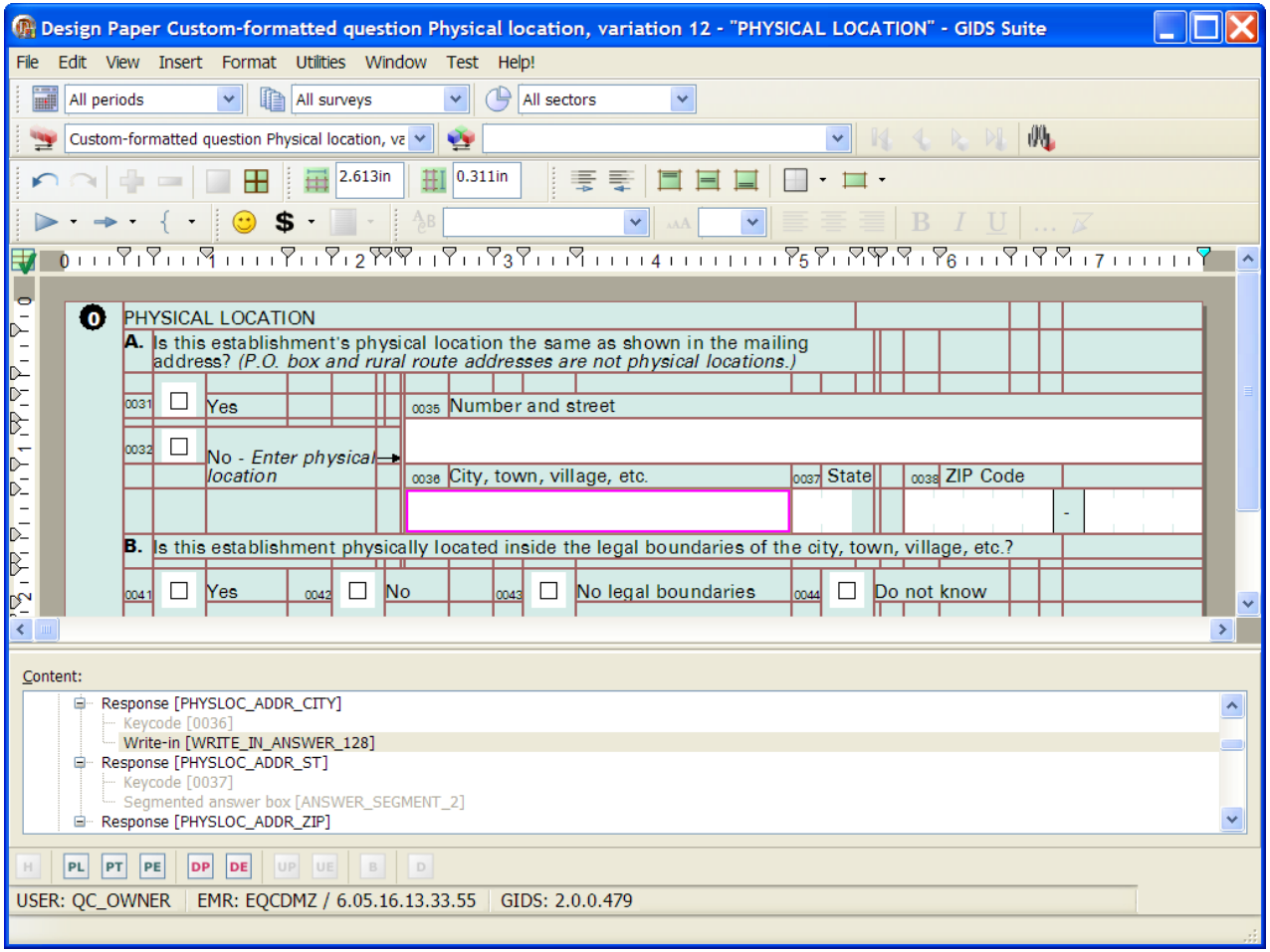
Economic Census

- Conducted every five years
- Surveys U.S. businesses on economic activity
- 3.5 million respondents
- ~800 questionnaires, average 12 pages each
- 2002 and 2007 Economic Census developed with “Generalized Instrument Design System”

GIDS Designer (1)

- Generalized Instrument Design System (GIDS) Designer
  - Connects to content metadata database
  - Designs paper and electronic questionnaire layouts
  - Designs electronic behavior (“edits”)
  - Supports reusable questions (reduced designed pages from ~10,000 to ~1,600)
  - Used for the 2002 and 2007 Economic Censuses
  - Used for annual economic surveys

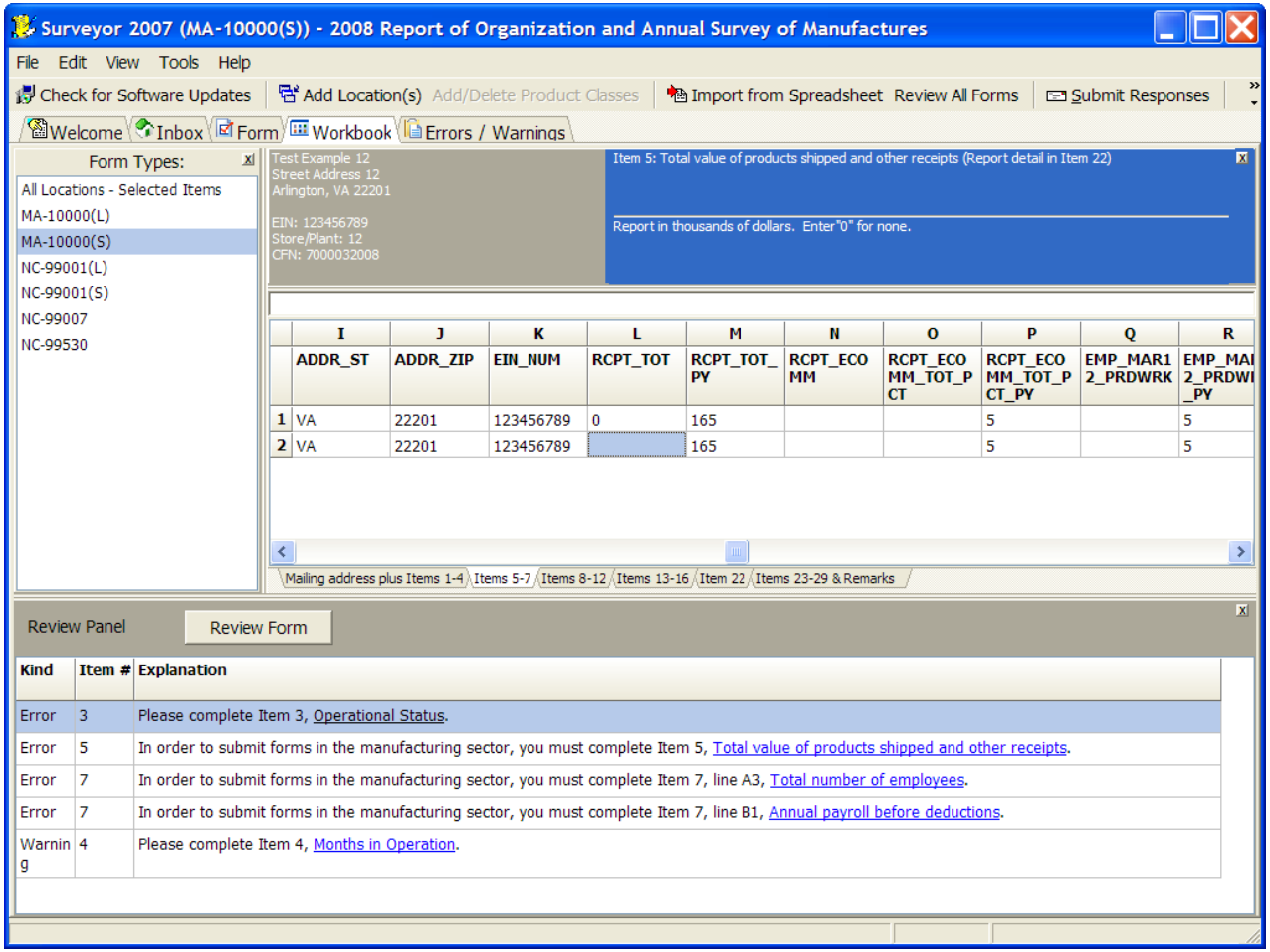
GIDS Designer (2)



GIDS Surveyor (1)

- Generalized Instrument Design System (GIDS) Surveyor
  - Computerized Self-Administered Questionnaire
  - Supports both form-view and spreadsheet-view
  - Validates responses through behavior “scripts”
  - Submits responses to servers through XML and web services
  - Respondents used GIDS Surveyor to provide 877,676 questionnaire responses in 2007

GIDS Surveyor (2)



Major Data Defect (1)

- Defect anecdote
  - We implement a performance enhancement in GIDS Surveyor in October 2008
  - Performance enhancement speeds loading of response data through “pre-fetch” algorithm that assumes a single questionnaire has no more than 10,000 responses areas
  - On December 20th, we receive reports of data loss in very large response data sets
  - We release a fix on December 22nd

Major Data Defect (2)

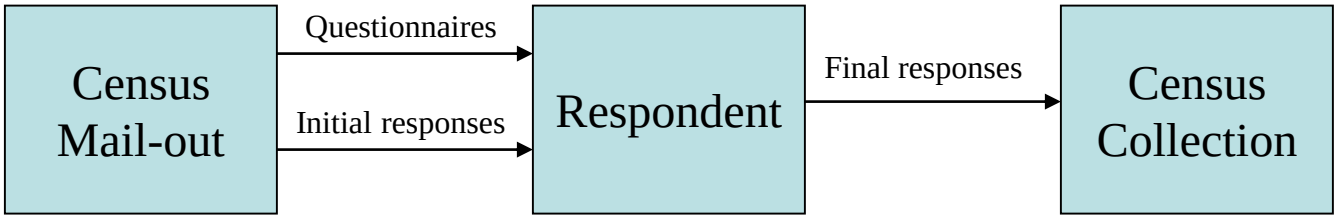
- Cause & Effect
  - Some “unusual” questionnaires had more than 10,000 response data elements
  - Responses “safe” in the local response database, just not included in the XML response file
  - Resubmission with updated software fixes the problem
- Lessons Learned
  - By necessity, performance enhancement often occurs late in the development cycle, leaving less time to uncover defects
  - In our experience, data defects that escape “to the wild” are often caused by performance enhancements

Minor Data Defect (1)

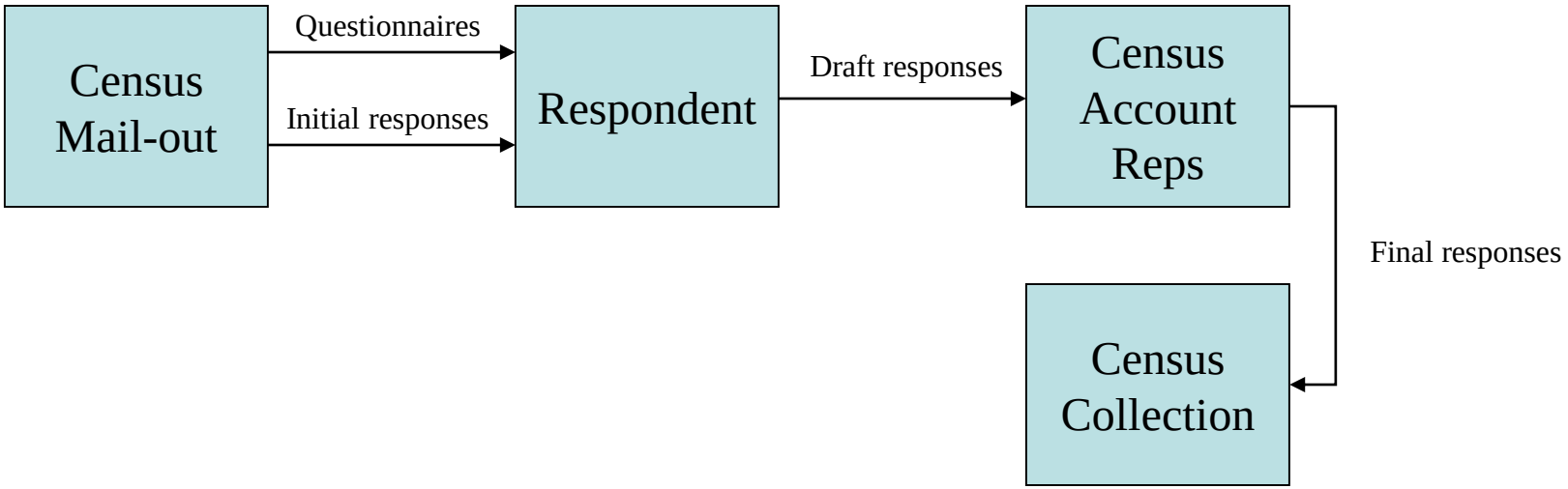
- Defect anecdote
  - On December 20th, we receive reports of incorrect metadata for certain cases
  - We determine that the defect occurs when Surveyor is used in an unexpected operational context
  - We release a fix on December 26th

Minor Data Defect (2)

Initial operational context



"Real World" operational context



Minor Data Defect (3)

- Cause & Effect
  - We recorded metadata to indicate responses were “final” too early in the process in the new operational context
  - Downstream data processing could have “massaged” the response data
  - We did release a fix to Surveyor
- Lessons Learned
  - Enumerate and thoroughly test all operational scenarios
  - Human nature focuses on the “normal” cases, ignores rare events (e.g., Nassim Taleb’s “Black Swan” events)

Process Data Defect (1)

- Defect Anecdote
  - Census sends questionnaires with “pre-listed” responses based on prior period reporting
  - Surveyor maintains a “snapshot” of the original “pre-listed” responses in the response database
  - Surveyor uses several approaches to minimize XML response data set file size for transfer
  - We received report that in some cases, Surveyor was omitting responses incorrectly

Process Data Defect (2)

Data Element	Original Value	Current Value
Company Name	“Fenestra Technologies”	(empty)
Contact Name	(empty)	“Rick Rogers”
Contact Phone	(empty)	(empty)

- Original (defective) algorithm
  - If Current Value is empty, do not write response data
- Corrected algorithm
  - If both Original Value and Current Value are empty, do not write response data

Process Data Defect (3)

- Cause & Effect
  - Caused by an optimization in boundary case
  - Discovered through extensive regression testing
  - Rare: 12 cases out of several million
- Lessons Learned
  - Many defects occur in boundary or “edge” cases
  - Process data defects can be subtle
  - Sometimes “no fix” is “best fix”

Lessons Learned

- Beware performance optimizations
- Foreseeing boundary cases is tough:
  - Rely on experience: use check-lists
  - Consider testing (or simulating) with random data
- Carefully impact of options for fixes:
  - Upstream, in data collection
  - Downstream, in data processing
  - No fix at all