Technomics
Better Decisions Faster

The costverse for the FlexFile

Ben Berkman and Justin Cooper

August 12, 2020

2020 ICEAA paper: "The FlexFile Framework: Preparing for Powerful Analysis in R"

Today's Presenters

Technomics Innovation Lab

Ben Berkman

Ben is a lead analyst at Technomics. He has supported OSD CAPE's Defense Cost and Resource Center since 2017. A lead of CAPE's FlexFile initiative, he is heavily involved in building workflows in R to support the FlexFile submission process. He was part of the CADE team to win this year's ICEAA Educator of the Year award. He is an ICEAA certified analyst and holds a BS in Economics from Penn State University (2017).

Justin Cooper

Justin is a senior associate at Technomics. In addition to supporting OSD CAPE's Defense Cost and Resource Center since 2018, he has also been involved in supporting CAPE O&S analytics tools working in applications such as R, Excel, and Tableau. He was named this year's ICEAA Junior Analyst of the Year. He holds a BS in Economics from Penn State University (2018).

Agenda

- The basics
  - What is the FlexFile?
  - Why use R?
  - Why use R with the FlexFile?
  - Why use R packages?
- The **costverse**
  - Packages
  - Functions
- Demo
  - The **costverse** in action
- Want to learn more?
- Questions and discussion

What is the FlexFile?

"Legacy" DD 1921 Reports

- ❑ Work Breakdown Structure
- ❑ Nonrecurring and Recurring
- ❑ Standard Functional Categories

FlexFile

- ❑ All DD 1921 Data Elements
- ❑ Monthly Time-phased Data
- ❑ "Account" Level Detail
- ❑ Contractor Functional Categories

Data Group A
Report Metadata
Approved Plan Number
Submission Event
Period of Performance
Reporting Organization
As of Date
Date Prepared
more in the DID...

Data Group B
DD Form 2794 Data Elements
WBS Element
Order/Lot
End Item

Data Group C
Definitions and Remarks
WBS Element Definitions
Remarks by WBS Element
Summary Remarks

Data Group D
Summary Elements
Order/Lot
Subtotal
General and Administrative
Undistributed Budget
Management Reserve
Facilities Capital Cost of Money
Contract Fee
Contract Price

Data Group E
Actuals To Date (ATD)
Account
Reporting Period
CLIN
Nonrecurring or Recurring
Functional Category / Overhead
Standard Functional Category
Unit/Sublot
WBS Element
Order/Lot and End Item
ATD (Dollars and Labor Hours)

Data Group F
Allocation Methodology
Allocation Method

Data Group G
Forecasts At Completion (FAC)
FAC (Dollars)
FAC (Labor Hours)

Greatest value to cost estimator, in many cases, will be the additional insight requested in Group E

Legacy Element

The FlexFile combines both the old and the new in one report that supports both top-down and bottom-up estimates

Berkman, Benjamin and Marc Stephenson. The FlexFile: The Next Generation in Contractor Cost Data Reporting. *ICEEA Professional Development & Training Workshop*, Tampa, FL, May 14-17, 2019. Unpublished Conference Paper. Online.

What R We Talking About?

- Excel is often the tool of choice for cost analysts, but has several downsides:
  - Lack of consistency
  - Lack of object language
  - Lack of statistical features
  - Lack of scalability
  - Lack of graphical features
  - Fixed state

- R provides an open-source environment with fast and user-friendly packages
  - tidyverse: popular for data manipulation and visualization
  - Logical and well defined approach

Import → Tidy → Transform → Visualise → Model → Communicate

Understand

Program
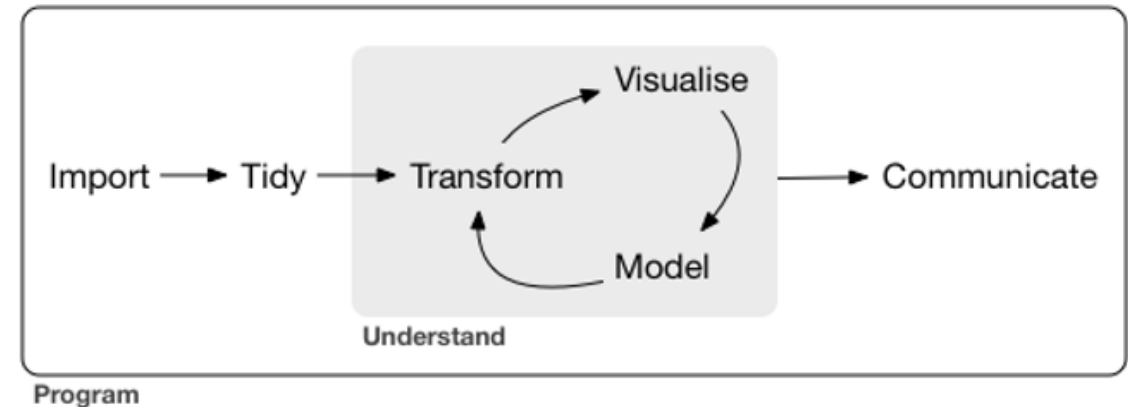
"…a collection of R packages that share a high-level design philosophy and low-level grammar and data structures."

- Hadley Wickham, Chief Scientist at RStudio

Wickham, H., et al. 2019. "Welcome to the Tidyverse." The Comprehensive R Archive Network. https://cran.r-project.org/web/packages/tidyverse/vignettes/paper.html

Why Use R with the FlexFile?

- tidyverse provides general framework for efficiently working with data

- Necessary repetitive and tedious tasks when working with a custom data format such as the FlexFile still exist

- R solution provides a consistent, scalable, and dynamic method to import, tidy, transform, and visualize the FlexFile

Why Organize Code in Packages?

How do we bring multiple FlexFiles into an R data frame?

Inefficient…

Taking the necessary steps to manually read 15+ JSON files into R, join 10+ lookup tables, apply allocations, add a unique ID for each FlexFile, repeat for each FlexFile, then manually stack the files into one data frame.
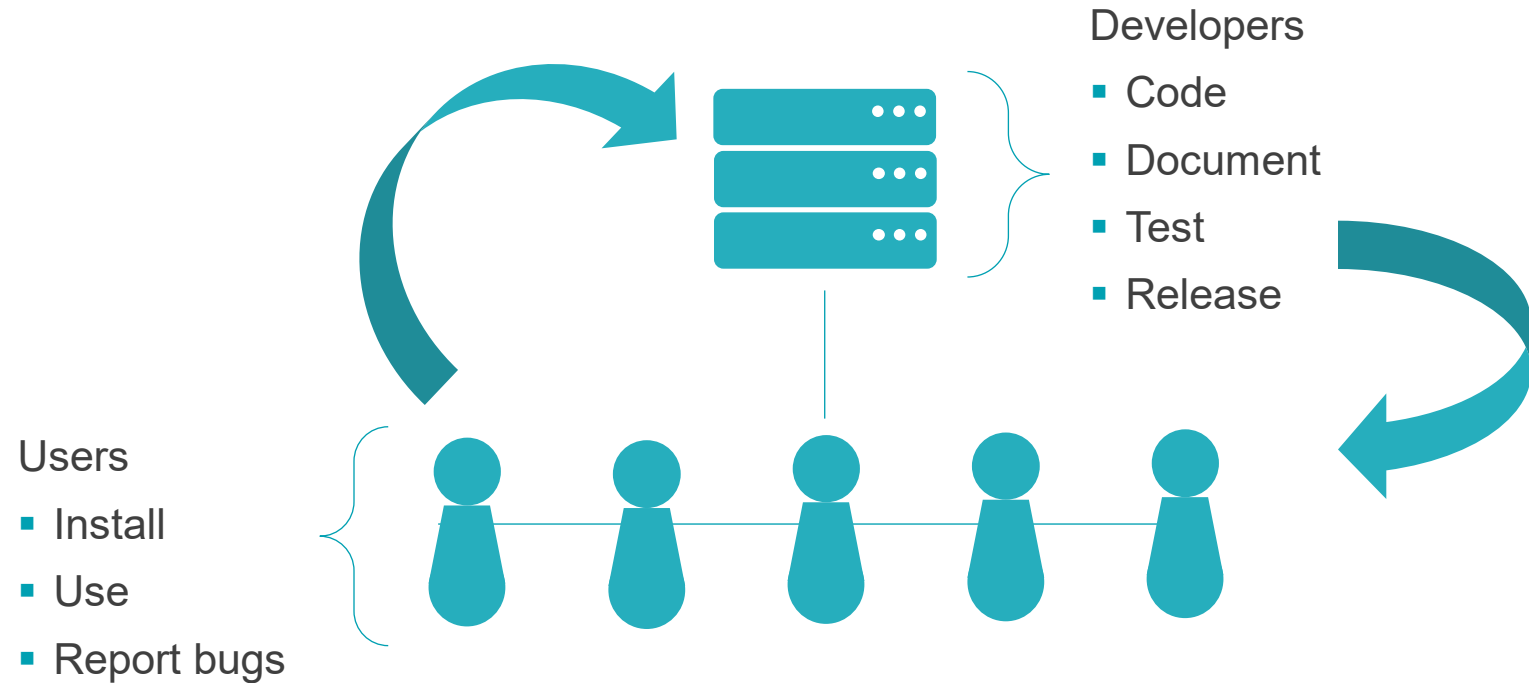
Efficient…

Simply running:

```
read_folder(folder, read_ff) %>%
        list_index_to_col() %>%
        stack_ff() %>%
        allocate_ff() %>%
        flatten_ff()
```

Why Organize Code in Packages?

- Developers build custom for you. Test and create a stabilized, working version of a package.

- Shareable code developed and delivered by authoritative R users

- Prior versions saved and archived via version control

Developers
- Code
- Document
- Test
- Release
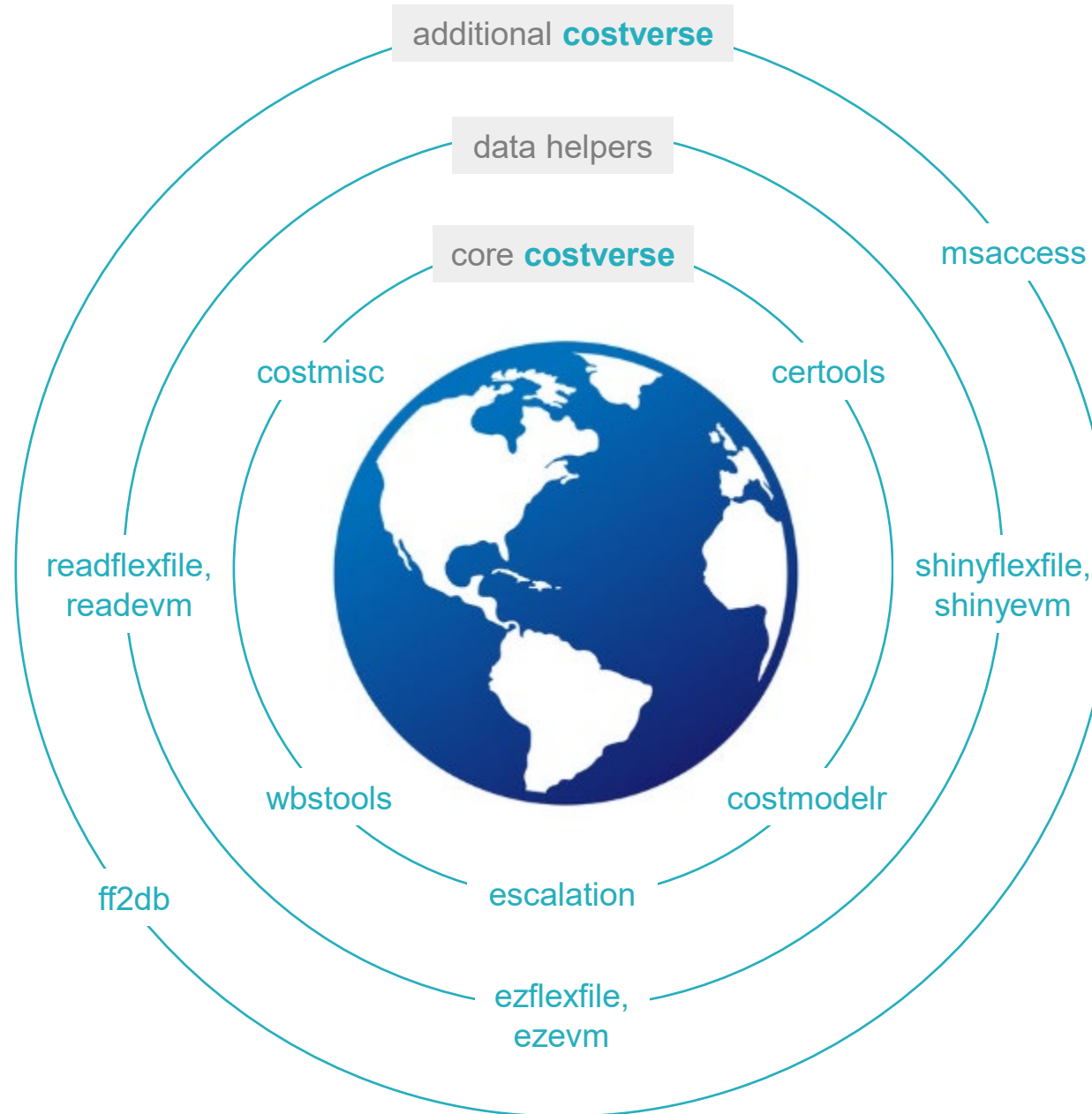
Users
- Install
- Use
- Report bugs

Packages provide the most efficient way to share, document, and update code between users

8

The **costverse** universe

additional **costverse**

data helpers

core **costverse**

msaccess

costmisc

certools

readflexfile, readevm

shinyflexfile, shinyevm

wbstools

costmodelr

ff2db

escalation

ezflexfile, ezevm

The **costverse** for the FlexFile

readflexfile
- read data into R
- tidy data into flat file

ezflexfile
- summary tables
- exploratory plots

shinyflexfile
- interactive dashboard

- Three FlexFile-specific packages intended to build on each other
- Mirrors tidyverse philosophy and syntax
- Functions for both FlexFile and Quantity Data Report

The **costverse[readflexfile]**

readflexfile
- **read data into R**
- **tidy data into flat file**

ezflexfile
- **summary tables**
- **exploratory plots**

shinyflexfile
- **interactive dashboard**

readflexfile functions:

- ▪ **read_ff**: Read one FlexFile into a list of tibbles
- ▪ **stack_ff**: Stack list of multiple FlexFile submissions into one list of tibbles
- ▪ **flatten_ff**, **flatten_qdr**: Flatten FlexFile and Quantity Data report
- ▪ **allocate_ff**: Apply allocation methodologies provided

additional helper functions exported from **costmisc** package:

- ▪ **read_folder**: Read a folder of files. Supports reading multiple FlexFiles.
- ▪ **add_id_col**: Add ID column to data frame. Support reading multiple FlexFiles.
- ▪ **list_index_to_col**: Add ID column to data frame. Support reading multiple FlexFiles.

The **costverse[ezflexfile]**

Technomics
Innovation Lab

readflexfile
- read data into R
- tidy data into flat file

ezflexfile
- **summary tables**
- **exploratory plots**

shinyflexfile
- interactive dashboard

ezflexfile functions:

- **tbl_[functions]**: Summarize FlexFile data
- **plot_ff**: Plot FlexFile data
- **plot_bar**: Plot FlexFile bar charts
- **plot_area**: Plot FlexFile over time
- **plot_sankey**: Plot interactive FlexFile Sankey diagram
- **plot_sunburst**: Plot interactive FlexFile Sunburst charts

The **costverse[shinyflexfile]**

readflexfile
- read data into R
- tidy data into flat file

ezflexfile
- summary tables
- exploratory plots

shinyflexfile
- **interactive dashboard**

shinyflexfile functions:

▪ **shiny_ff**: Launch a Shiny dashboard application to analyze the FlexFile

Demo

Want to learn more?

Adam James

Ben Berkman

Justin Cooper

Adam, Ben, and Justin created the costverse packages for the FlexFile. Together, they bring more than a decade of experience in cost analysis, R tool development, and client support.

✉ **ajames@Technomics.net**

✉ **bberkman@Technomics.net**

✉ **jcooper@Technomics.net**

PROFESSIONAL DEVELOPMENT & TRAINING WORKSHOP

MAY 18-20, 2021 • MINNEAPOLIS

ICEAA

Abstract Summaries Due November 3, 2020

iceaaonline.com/cfp2021