

Getting a New Generation to a New Normal on the Mainframe

Reg Harbeck
Mainframe Analytics

August 14, 2013
Session Number 14177

My Background

- B.Sc. in Computer Science
- 25 years in IT and mainframe
- Worked for private industry, then government, then CA
- In 2005 introduced Mainframe Continuity Planning as whitepaper and SHARE presentation
- Co-founded Mainframe Analytics in 2011
 - <http://MainframeAnalytics.com>
 - <http://MainframeAnalytics.blogspot.com>

Who Are You?

- Highly Experienced Mainframer (30+ years)
- Experienced Mainframer (15 – 29 years)
- Trusted Mainframer (5-14 years)
- New Mainframer (0 – 4 years)
- Non-mainframe IT professional
- Potential Mainframer
- Ex-Mainframer
- IT management
- Curious
- Mainframe skeptic
- Mainframe retiree

It Happened

- Your best mainframers began to retire
- You couldn't leave the mainframe
- You hired newbies, repurposed non-mainframers, and contracted retirees
- And everything turned out OK!
- Here's what you did right

Takeaways

1. Cherish Your Legacy
2. Build on What Works
3. Blaze a Trail

Cherishing Your Legacy

- Legacy: “it works”



1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

Cherishing Your Legacy

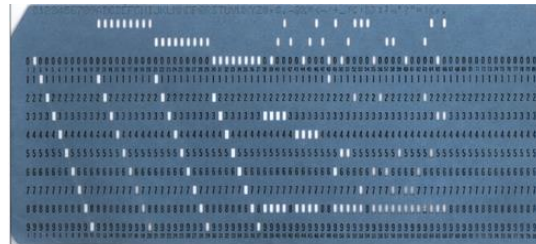
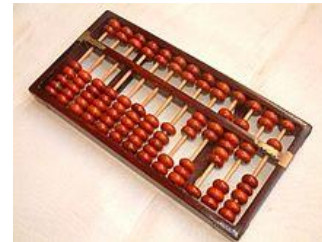
- Mainframe: the platform created by a culture



1. Cherish Your Legacy
2. Build on What Works
3. Blaze a Trail

History

- Stonehenge
- Abacus
- Babbage's difference engine
- Punch cards, electric tabulators, IBM



1. Cherish Your Legacy
2. Build on What Works
3. Blaze a Trail

History

- World War II
 - Ballistics
 - Code making and breaking
 - Enigma machine
 - Alan Turing



1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

Early Computers

- Computer: The first job title to be automated
- Business, academic and military technologies
- IBM



1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

Early Computers

- IBM 701
- SHARE founded in 1955
- Assembler, PL/1, Fortran, COBOL
- Grace Hopper



1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

April 7, 1964: System/360

- After nine years' gestation, IBM and SHARE had a love child



1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

Getting the Bugs Out

- OS/360, DOS
- Software as a product
- Databases
- Security
- CICS
- Statement of Integrity
- VM
- Backup and Recovery
- Automation
- Systems Management
- Practice, practice, practice

1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

Building the Culture

- Able, Baker, Charlie, Dog, Easy, Fox
- Military, Academia, Big Business
- Plan, test, allocate
- Life cycle management, change control
- Capacity planning, chargeback
- Backup and recovery
- 24 x 7 support, pagers, 2400 baud terminals

1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

Dinosaur?

- 1980's: Departmental and petty cash computers
- Teenage mainframe suddenly a “dinosaur”
- Other platforms grew up from single-user OSES
 - Needed to be “lite”
- Unix: 1969 (44 years old)
- MS-DOS: 1981 (32 years old)
- Macintosh: 1984 (29 years old)
- Windows: 1985 (28 years old)
- Linux: 1991 (22 years old)

1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

Dark Night of the Sole [Functional Platform]

- Mainframe lost its sizzle, but more was at stake
- Non-mainframe platforms tried to take on more
 - Generally, just got simple or new workloads
 - Often back-ended by mainframe
- Mainframe careers lost glamour
- Few new mainframers
- The mainframe kept running

1. **Cherish Your Legacy**
2. Build on What Works
3. Blaze a Trail

Y2K

- Fixed – too well?
- Billions of lines of code that now truly work



1. Cherish Your Legacy
2. Build on What Works
3. Blaze a Trail

That Ain't Just Data!

- It's time to radically care about the mainframe
 - Careers, businesses, the economy rely on this
 - Successfully!
 - Ask yourself: "What in my life do I care deeply about and why?"
 - Caring leads to sharing: let other people know about the platform you trust!
 - Decision makers and influencers
 - Non-IT acquaintances at coffee or cocktails or golf
 - Place of worship, Toastmasters, Rotary, Mensa...
 - Seriously! This really matters!
1. **Cherish Your Legacy**
 2. Build on What Works
 3. Blaze a Trail

What Works?

- Something old
 - Tried and proven
 - Hardware
 - Operating Systems
 - Systems management software
 - Databases, languages
 - Applications
 - Scrupulous practices, culture

1. Cherish Your Legacy
- 2. Build on What Works**
3. Blaze a Trail

What Works?

- Something new
 - Latest technologies
 - zEnterprise
 - Software innovations
 - Ease of installation



1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

What Works?

- Something borrowed
 - The best of other platforms!
 - TCP/IP
 - Browser-based interfaces
 - Graphical management workspaces
 - Enterprise monitoring, automation, management

1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

What Works?

- Something Blue



1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

What Doesn't Work?

- Obsolete systems programs
 - Often in assembler
 - Rarely well-documented
 - Multiple layers of modifications
 - Original authors often long gone
 - Vendor software now often does the same thing, only better
 - And it's already running on your mainframe

1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

What Doesn't Work?

- Out-of-date configurations
 - “Set and forget” gets stale after a few decades
 - Often new options use fewer resources
 - Replace those obsolete in-house programs
 - Your business needs have likely changed

1. Cherish Your Legacy
- 2. Build on What Works**
3. Blaze a Trail

What Doesn't Work?

- Overlapping vendor software
 - May no longer meet a business need
 - Possible duplication of functionality

1. Cherish Your Legacy
- 2. Build on What Works**
3. Blaze a Trail

What Doesn't Work?

- Obsolete attitudes
 - Mutual condescension between platforms
 - Reluctance to let new people learn by experience
 - Rigid responses to innovative ideas

1. Cherish Your Legacy
- 2. Build on What Works**
3. Blaze a Trail

What Doesn't Work?

- Bad apples
 - It's easy to forget they're mostly long gone
 - Other platforms still have lots
 - With a new cohort of mainframers, there'll be more
 - If your security, auditing, change control, etc. aren't tight, you're in for unpleasant surprises
 - Watch out for loose ends like extraneous APF lib's



1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

How to Build?

- Start with the people you already have
 - Value their knowledge and experience
 - Make sure they know you do
 - Encourage them to begin mentoring
 - Slowly hand over their responsibilities
 - They won't run out of work
 - Look into some post-retirement contracting

1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

How to Build?

- Add new people
 - Repurposed non-mainframe IT staff
 - New kids
 - Experienced mainframers from other organizations
 - Disabused un-disillusioned ex-mainframers

1. Cherish Your Legacy
- 2. Build on What Works**
3. Blaze a Trail

Build with Your New People

- Send them on courses (in-person and CBT)
- Have them map what they learned to your environment and report on it
- Send them on conferences
- Get them to sign up for zNextGen
- Encourage them to learn the culture
- Make sure they each have at least one mentor



1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

Build with Your New People

- Assign them projects to learn and document your environment
 - What in-house code is there?
 - What's obscure?
 - What's obsolete?
 - Who still understands it?

1. Cherish Your Legacy
- 2. Build on What Works**
3. Blaze a Trail

Build with Your New People

- Have them learn and document your culture
 - Change Control, DR, Capacity Planning, Security...
 - Why, how, who?
 - Encourage them to ask challenging questions and not give up until they have a satisfactory answer
 - Have them suggest alternative ways to get the same or better results
1. Cherish Your Legacy
 2. **Build on What Works**
 3. Blaze a Trail

Build with Your New People

- Get to know the capabilities of your vendor software
 - Start with the operating system
 - Then review other vendor software
 - How is it currently used and configured?
 - What new features exist?
 - How can it be run more optimally?
 - What in-house code can it supersede?
 - What new business needs can it respond to?
1. Cherish Your Legacy
 2. **Build on What Works**
 3. Blaze a Trail

Use a Taxonomy

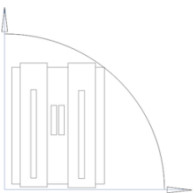
- To understand your software, categorize it by “what” and “why”
- Don’t assume its original purpose or marketing message is the value it brings now
- Many solutions have multiple values
- One approach is the Mainframe Analytics Taxonomy
- See <http://mainframeanalytics.blogspot.com> for elaboration

1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

Mainframe Analytics Taxonomy: Behavior (“What”) Dimension

1. Data Handling
2. Device Interfacing
3. Applications and Automation
4. Context Management
5. Optimization
6. Quality and Lifecycle
7. Production

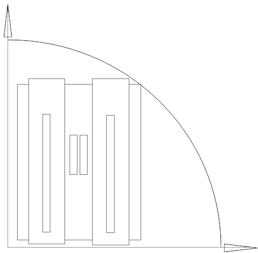
1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail



Complete your sessions evaluation online at SHARE.org/BostonEval

Mainframe Analytics Taxonomy: Business Need (“Why”) Dimension

- A. Business Enablement
- B. Continuity
- C. Security, Integrity and Compliance
- D. Cost-Effective Operations
- E. Analysis and Planning
- F. New Business Value



1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

Build with Your New People

- Assign them to present suggested improvements
 - First to peers and open-minded supporters
 - Then to experienced people and management
 - If they can project a real business benefit, give them a chance
 - Reduced CPU and other resource usage
 - Reduced software, hardware, etc. costs
 - Reduced maintenance effort
 - Sometimes, just learning can be enough of a benefit
1. Cherish Your Legacy
 2. **Build on What Works**
 3. Blaze a Trail

Implementing Suggestions

- Use the opportunity to get into good, scrupulous habits
 - Document, document, document
 - Plan, review, fine-tune
 - Test, test, test
 - Full change control, backout plan
 - Present on experience, retain lessons learned

1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

Keep Doing It

- A true mainframer is a professional
 - Professionals keep learning all their lives
 - There's no limit to the number of ways to beneficially improve
 - Get your new people in the habit of finding, suggesting and making improvements
 - Recognize and reward successes
 - Learn from failures and reward good presentations of lessons learned
1. Cherish Your Legacy
 2. **Build on What Works**
 3. Blaze a Trail

Let the Whole World Know!

- As the mainframe, which already works, keeps getting better, don't hide it
- Write articles for in-house publications
- Give lunch'n'learn presentations about what you do and learn
- Let other parts of your organization know
- Present at conferences, write articles for industry publications
- Advance your career!
 1. Cherish Your Legacy
 2. **Build on What Works**
 3. Blaze a Trail

What About the Experienced?

- Give 'em some WIIFM
- Respect your mentors
- Give them credit in presentations and articles
- Encourage socializing across generations
- Take their sensibilities seriously
- When they retire, treat it like a graduation
- Find ways to include them after if they wish

1. Cherish Your Legacy
2. **Build on What Works**
3. Blaze a Trail

Blaze a Trail!

- You now have a whole stable full of mainframe evangelists
- They already know other platforms from their pre-mainframe days
- They've learned your mainframe environment and business needs
- They're the next wave in responsible computing – turn 'em loose!

1. Cherish Your Legacy
2. Build on What Works
3. **Blaze a Trail**

The Return of Responsibility

- Responsible computing doesn't require mainframes – but it helps
- Encourage cross-pollination of mainframe strengths and non-mainframe innovations
- Get your enthusiasts to suggest IT improvements that go beyond the mainframe
- Become known for being the platform and people of innovation

1. Cherish Your Legacy
2. Build on What Works
3. **Blaze a Trail**

Question Accepted Wisdom

- Hold forums for discussion of responsible computing
- Ask hard questions with no easy answers
- What's sizzle, what's steak, and what's waste?
- Why does the mainframe do DR, Capacity Planning, Change Control, etc.?
- When can/should other platforms?
- If they can't, what workloads should they forfeit?

1. Cherish Your Legacy
2. Build on What Works
3. **Blaze a Trail**

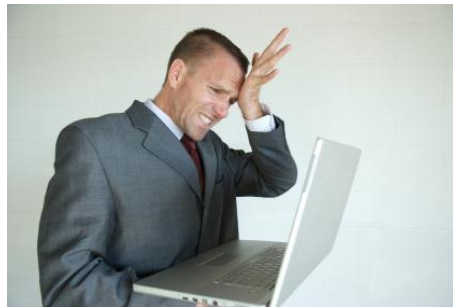
Ask For Work

- Mainframers are seen as resistant to new applications, initiatives
- We get approached and tell people they'll have to wait
- Instead, figure out how to go after new business and properly staff and implement it
- Market your strengths, then show them
- Don't wait to be asked

1. Cherish Your Legacy
2. Build on What Works
3. **Blaze a Trail**

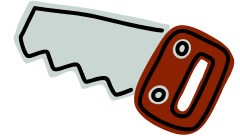
Coulda Done it on a Mainframe!

- When chatting about failed or insufficiently successful non-mainframe projects, pay attention
- If mainframe attitudes and technology could have prevented it, don't be shy
- “Shame they didn't do it the mainframe way”
- Mean it, and be prepared to defend it



1. Cherish Your Legacy
2. Build on What Works
3. **Blaze a Trail**

Sharpen Your Saw



- Never lose sight of advancing your mainframe environment
- Find a way to keep, maintain, and show your learnings
 - Consider a wiki
- Keep the doors open for new suggestions and innovations
- Don't pooh-pooh non-mf'ers

1. Cherish Your Legacy
2. Build on What Works
3. **Blaze a Trail**

Celebrate!

- Never forget: it's all human!
- “It’s just business” is B.S.
- “We’re only human” is better
- Nothing succeeds like success – don’t hide it!
- Nerds just wanna have fun!



1. Cherish Your Legacy
2. Build on What Works
3. **Blaze a Trail**

How You Won

1. Cherish Your Legacy
2. Build on What Works
3. Blaze a Trail

