Managing FOSS for Business Results

CJ Fearnley
cjf@LinuxForce.net

President & CEO
LinuxForce, Inc.
http://www.LinuxForce.net

29 October 2009

Presentation to LinuxForce Seminar on
Managing Free and Open Source Software (FOSS) for Business Results
at The Franklin Institute, Fifth Floor Conference Center, Philadelphia, PA

http://www.CJFearnley.com/managing.foss.2009.pdf

Managing FOSS for Business Results

Software Management

Software Maintenance Lifecycle

- Needs Assessment / Requirements Analysis
- Research / Testing / Evaluation / Development (all software is “new” due to exponential trends)
- Installation / deployment
- Integration
- Monitoring
- Security upgrades
- Support
- New release upgrades

Software Management

Software Maintenance Lifecycle

- Needs Assessment / Requirements Analysis
- Research / Testing / Evaluation / Development (all software is “new” due to exponential trends)
- Installation / deployment
- Integration
- Monitoring
- Security upgrades
- Support
- New release upgrades

Software Management

Software Maintenance Lifecycle

- Needs Assessment / Requirements Analysis
- Research / Testing / Evaluation / Development (all software is “new” due to exponential trends)
- Installation / deployment
- Integration
- Monitoring
- Security upgrades
- Support
- New release upgrades

Software Management

Software Maintenance Lifecycle

- Needs Assessment / Requirements Analysis
- Research / Testing / Evaluation / Development (all software is “new” due to exponential trends)
- Installation / deployment
- Integration
- Monitoring
- Security upgrades
- Support
- New release upgrades

Managing Upgrades with FOSS

- In the FOSS world, you almost never need to upgrade hardware to take advantage of new software versions.
- In the FOSS world, community support protocols require upgrading to either the newest version or to a version supported by your distribution. The timeframe for supported software is sometimes shorter than for commercial software, but the benefits of bug fixes and new features come with no new licensing costs AND most FOSS packages and the distributions make upgrading easy.
- With Debian / Ubuntu upgrades are almost painless!
- Risks with upgrades
  - New features may require staff re-training
  - There may be glitches that require support
  - Near zero downtime upgrades require careful planning

Managing Upgrades with FOSS

- In the FOSS world, you almost never need to upgrade hardware to take advantage of new software versions.
- In the FOSS world, community support protocols require upgrading to either the newest version or to a version supported by your distribution. The timeframe for supported software is sometimes shorter than for commercial software, but the benefits of bug fixes and new features come with no new licensing costs AND most FOSS packages and the distributions make upgrading easy.
- With Debian / Ubuntu upgrades are almost painless!
- Risks with upgrades
  - New features may require staff re-training
  - There may be glitches that require support
  - Near zero downtime upgrades require careful planning

Managing Upgrades with FOSS

- In the FOSS world, you almost never need to upgrade hardware to take advantage of new software versions.
- In the FOSS world, community support protocols require upgrading to either the newest version or to a version supported by your distribution. The timeframe for supported software is sometimes shorter than for commercial software, but the benefits of bug fixes and new features come with no new licensing costs AND most FOSS packages and the distributions make upgrading easy.
- With Debian / Ubuntu upgrades are almost painless!
- Risks with upgrades
  - New features may require staff re-training
  - There may be glitches that require support
  - Near zero downtime upgrades require careful planning

Managing Upgrades with FOSS

- In the FOSS world, you almost never need to upgrade hardware to take advantage of new software versions.
- In the FOSS world, community support protocols require upgrading to either the newest version or to a version supported by your distribution. The timeframe for supported software is sometimes shorter than for commercial software, but the benefits of bug fixes and new features come with no new licensing costs AND most FOSS packages and the distributions make upgrading easy.
- With Debian / Ubuntu upgrades are almost painless!
- Risks with upgrades
  - New features may require staff re-training
  - There may be glitches that require support
  - Near zero downtime upgrades require careful planning

A Winning Formula for FOSS: Upgradeability

- Especially with Debian & Ubuntu, FOSS can be routinely upgraded so that bug fix and new feature upgrades stream to your business like interest payments.
- Easy upgradeability makes the software lifecycle more like an ecosystem that continually changes in a slow gradual way.

Comprehensive Anticipatory Systems Administration

- Comprehensive because what isn't considered is what will "get ya"
- Anticipatory because what isn't planned for will "get ya"
- This concept is derived from Buckminster Fuller's Comprehensive, Anticipatory Design Science
- R. Buckminster "Bucky" Fuller (July 12, 1895 - July 1, 1983) was an American architect, author, designer, futurist, inventor, and visionary.
- It provides LinuxForce with a philosophy of systems administration to inspire our team to address issues that are sometimes overlooked

Comprehensive Anticipatory Systems Administration

- Comprehensive because what isn't considered is what will "get ya"
- Anticipatory because what isn't planned for will "get ya"
- This concept is derived from Buckminster Fuller's Comprehensive, Anticipatory Design Science
- R. Buckminster "Bucky" Fuller (July 12, 1895 - July 1, 1983) was an American architect, author, designer, futurist, inventor, and visionary.
- It provides LinuxForce with a philosophy of systems administration to inspire our team to address issues that are sometimes overlooked

Comprehensive Anticipatory Systems Administration

- Comprehensive because what isn't considered is what will "get ya"
- Anticipatory because what isn't planned for will "get ya"
- This concept is derived from Buckminster Fuller's Comprehensive, Anticipatory Design Science
- R. Buckminster "Bucky" Fuller (July 12, 1895 - July 1, 1983) was an American architect, author, designer, futurist, inventor, and visionary.
- It provides LinuxForce with a philosophy of systems administration to inspire our team to address issues that are sometimes overlooked

Notable Quotable

[Design Science is] the effective application of the principles of science to the conscious design of our total environment in order to help make the Earth's finite resources meet the needs of all humanity without disrupting the ecological processes of the planet.

— Buckminster Fuller

Notable Quotable

It is one of the self-disciplined responsibilities of comprehensive, anticipatory design science always to include fail-safe, automatically switched-in, alternate circuitry for mechanical functioning whenever a prime-function facility is found wanting.

— Buckminster Fuller, Synergetics 183.00

Eternally Regenerative Software Administration

- Eternal because upgradeability means through release after release after release
- Regenerative because "change is normal" and so the new will sometimes replace the old: the ecosystem is eternally regenerative (or so we hope), software should be the same!
- Design and management by humans is essential to provide this "eternal regenerativity"
- This notion is derived from Buckminster Fuller's concept of the purpose of humans in Universe "in support of the integrity of eternally regenerative Universe"

Eternally Regenerative Software Administration

- Eternal because upgradeability means through release after release after release
- Regenerative because "change is normal" and so the new will sometimes replace the old: the ecosystem is eternally regenerative (or so we hope), software should be the same!
- Design and management by humans is essential to provide this "eternal regenerativity"
- This notion is derived from Buckminster Fuller's concept of the purpose of humans in Universe "in support of the integrity of eternally regenerative Universe"

Eternally Regenerative Software Administration

- Eternal because upgradeability means through release after release after release
- Regenerative because "change is normal" and so the new will sometimes replace the old: the ecosystem is eternally regenerative (or so we hope), software should be the same!
- Design and management by humans is essential to provide this "eternal regenerativity"
- This notion is derived from Buckminster Fuller's concept of the purpose of humans in Universe "in support of the integrity of eternally regenerative Universe"

Eternally Regenerative Software Administration

- Eternal because upgradeability means through release after release after release
- Regenerative because "change is normal" and so the new will sometimes replace the old: the ecosystem is eternally regenerative (or so we hope), software should be the same!
- Design and management by humans is essential to provide this "eternal regenerativity"
- This notion is derived from Buckminster Fuller's concept of the purpose of humans in Universe "in support of the integrity of eternally regenerative Universe"

Notable Quotable

My own working assumption of why we are here is that we are here as local-Universe information-gatherers and that we are given access to the divine design principles so that we can therefrom objectively invent instruments and tools – e.g., the microscope and the telescope – with which to extend all sensorial inquiring regarding the rest of the to-the-naked-eye-invisible, micro-macro Universe, because human beings, tiny though we are, are here for all the local-Universe information-harvesting and cosmic-principle-discovering, objective tool-inventing, and local-environment-controlling as local Universe problem-solvers in support of the integrity of eternally regenerative Universe.

— Buckminster Fuller

Notable Quotable

My own working assumption of why we are here is that we are here as local-Universe information-gatherers and that we are given access to the divine design principles so that we can therefrom objectively invent instruments and tools – e.g., the microscope and the telescope – with which to extend all sensorial inquiring regarding the rest of the to-the-naked-eye-invisible, micro-macro Universe, because human beings, tiny though we are, are here for all the local-Universe information-harvesting and cosmic-principle-discovering, objective tool-inventing, and local-environment-controlling as local Universe problem-solvers in support of the integrity of eternally regenerative Universe.

— Buckminster Fuller

Notable Quotable

My own working assumption of why we are here is that we are here as local-Universe information-gatherers and that we are given access to the divine design principles so that we can therefrom objectively invent instruments and tools – e.g., the microscope and the telescope – with which to extend all sensorial inquiring regarding the rest of the to-the-naked-eye-invisible, micro-macro Universe, because human beings, tiny though we are, are here for all the local-Universe information-harvesting and cosmic-principle-discovering, objective tool-inventing, and local-environment-controlling as local Universe problem-solvers in support of the integrity of eternally regenerative Universe.

— Buckminster Fuller

Notable Quotable

My own working assumption of why we are here is that we are here as local-Universe information-gatherers and that we are given access to the divine design principles so that we can therefrom objectively invent instruments and tools – e.g., the microscope and the telescope – with which to extend all sensorial inquiring regarding the rest of the to-the-naked-eye-invisible, micro-macro Universe, because human beings, tiny though we are, are here for all the local-Universe information-harvesting and cosmic-principle-discovering, objective tool-inventing, and local-environment-controlling as local Universe problem-solvers in support of the integrity of eternally regenerative Universe.

— Buckminster Fuller

Notable Quotable

My own working assumption of why we are here is that we are here as local-Universe information-gatherers and that we are given access to the divine design principles so that we can therefrom objectively invent instruments and tools – e.g., the microscope and the telescope – with which to extend all sensorial inquiring regarding the rest of the to-the-naked-eye-invisible, micro-macro Universe, because human beings, tiny though we are, are here for all the local-Universe information-harvesting and cosmic-principle-discovering, objective tool-inventing, and local-environment-controlling as local Universe problem-solvers in support of the integrity of eternally regenerative Universe.

— Buckminster Fuller

Conclusion

- Managing FOSS requires consideration of the whole software lifecycle
- “Comprehensive Anticipatory Systems Administration” captures LinuxForce’s approach to addressing these lifecycle issues
- Smooth upgradeability is an emergent property of many of the mature FOSS packages
- FOSS Upgradeability enables a new vision of “Eternally Regenerative Software Administration” to deliver businesses steady results by allowing them to continually use ever-improving software with minimal costs
- The result: Remote Responder: diligent, responsible, comprehensive systems management and monitoring services

Conclusion

- Managing FOSS requires consideration of the whole software lifecycle
- “Comprehensive Anticipatory Systems Administration” captures LinuxForce’s approach to addressing these lifecycle issues
- Smooth upgradeability is an emergent property of many of the mature FOSS packages
- FOSS Upgradeability enables a new vision of “Eternally Regenerative Software Administration” to deliver businesses steady results by allowing them to continually use ever-improving software with minimal costs
- The result: Remote Responder: diligent, responsible, comprehensive systems management and monitoring services

Conclusion

- Managing FOSS requires consideration of the whole software lifecycle
- “Comprehensive Anticipatory Systems Administration” captures LinuxForce’s approach to addressing these lifecycle issues
- Smooth upgradeability is an emergent property of many of the mature FOSS packages
- FOSS Upgradeability enables a new vision of “Eternally Regenerative Software Administration” to deliver businesses steady results by allowing them to continually use ever-improving software with minimal costs
- The result: Remote Responder: diligent, responsible, comprehensive systems management and monitoring services

Thank You

Thank You!

Any Questions?

http://www.CJFearnley.com/managing.foss.2009.pdf