

APPENDIX D

IN-HOUSE DEVELOPMENT

CHECKLISTS

DEVELOPMENT PLANNING

- Assign/matrix personnel with Ada 95 training and experience to the project.

Reference: Chapter 3, page 28, 32, 34, 41, 42

- Use risk management technology early.

Reference: Chapter 3, page 41

- Gain awareness of other Ada 95 adoption efforts through information buying.

Reference: Chapter 3, page 41

- Invest in training and consulting.

Reference: Chapter 3, page 41

- Involve staff, contractors, and support contractors in the adoption.

Reference: Chapter 3, page 41

- Benchmark and evaluate the tools to assess readiness.

Reference: Chapter 3, page 23, 24, 25, 41

- Employ incremental transition strategies.

Reference: Chapter 3, page 23, 41

- Ensure that the project's success does not rest on the use of the new technology.

Reference: Chapter 3, page 41

- Use the new technology to enhance your organization's competitiveness.
Reference: Chapter 3, page 41
- Amortize the risk of adoption over several programs.
Reference: Chapter 3, page 41
- Make use of Ada-based cost estimation tools such as Ada-COCOMO, REVIC, SLIM, and SoftCost-Ada.
Reference: Chapter 3, page 19
- Use pilot project efforts to help calibrate cost estimation models.
Reference: Chapter 3, page 19
- Minimize the number of simultaneous new technologies adopted.
Reference: Chapter 3, page 19
- Use the transition to Ada 83 as an approximation for the transition to Ada 95 in cost and schedule planning.
Reference: Chapter 3, page 19
- Contact early adopters to obtain lessons learned in order to minimize risk.
Reference: Chapter 3, page 19
- Track status and availability of compilers from all vendors.
Reference: Chapter 3, page 23
- Track the DOD policy on the use of non-validated and validated compilers.
Reference: Chapter 3, page 23
- Track the status of support tools for Ada 95.
Reference: Chapter 3, page 23
- Conduct benchmarking activities to evaluate prospective compilers.
Reference: Chapter 3, page 23
- Conduct pilot projects to evaluate the use of Ada 95 compilers.
Reference: Chapter 3, page 23
- Consider the opportunity to incrementally adopt Ada 95.
Reference: Chapter 3, page 23
- Ensure a detailed understanding of the technical issues of upward compatibility.
Reference: Chapter 3, page 27

- Assess the degree of Ada 95 upward compatibility of all existing Ada 83 software.
Reference: Chapter 3, page 27
- Investigate the degree to which new Ada 95 tools will be able to switch between Ada 83 and Ada 95 modes.
Reference: Chapter 3, page 27
- Evaluate the use of non-Ada legacy software, using technology such as “wrappers” and re-engineering to ensure cost-effective use of legacy code.
Reference: Chapter 3, page 27
- Ensure early and adequate training is provided for the staff.
Reference: Chapter 3, page 32
- Acquire and disseminate technology transfer materials to staff.
Reference: Chapter 3, page 32
- Seed trained people onto pilot projects.
Reference: Chapter 3, page 32
- Seed pilot project personnel onto full projects.
Reference: Chapter 3, page 32
- Ensure that multiple software development methods are not being used on the project.
Reference: Chapter 3, page 35
- Coordinate with CASE tool vendors to ensure that the tools support Ada 95 in the manner and timeframe appropriate to the project’s needs.
Reference: Chapter 3, page 35
- Ensure that staff is properly trained in an appropriate software development method.
Reference: Chapter 3, page 35
- Hire a consultant to mentor the project on new software development methods if they are being used for the first time.
Reference: Chapter 3, page 35
- Review the project to ensure that only the language or the development method has changed, but not both.
Reference: Chapter 3, page 35
- If both language and development method must change simultaneously, conduct a pilot project to evaluate the probability of success.
Reference: Chapter 3, page 35

- ❑ Employ a SETA contractor to monitor technical risks during development.
Reference: Chapter 3, page 35
- ❑ Identify all applicable standards, COTS, and legacy software.
Reference: Chapter 3, page 38
- ❑ Determine the project's bindings needs and risk areas.
Reference: Chapter 3, page 38
- ❑ Investigate the current wrapper technology to support the continued use of legacy code during system migration.
Reference: Chapter 3, page 38
- ❑ Ensure that the project's architecture is mapped to an "open system" standard such as the NIST APP or DISA's TAFIM.
Reference: Chapter 3, page 38

**SOFTWARE
CREATION**

- Provide instruction for tailoring Software Development Plans.
Reference: Chapter 4, page [53](#)
- Determine whether the development tool set supports all language features needed.
Reference: Chapter 3, page [23](#), [41](#)
- Evaluate the in-house staff's approach to using wrapper technology to preserve legacy code during system upgrade or migration.
Reference: Chapter 3, page [28](#), [31](#), [38](#)
- Evaluate staff's approach to code reuse (Ada and other languages).
Reference: Chapter 3, page [20](#)
- Make use of reuse code written in Ada and other languages.
Reference: Chapter 3, page [20](#)
- Make use of existing legacy code within new Ada 95 code if applicable.
Reference: Chapter 3, page [38](#)

**Post-
Deployment
Software
Support**

- Assess the adoption risk if proposed by in-house staff during the project.
Reference: Chapter 3, page 41
- Update formal software review procedures to include evaluation of upward compatibility of current Ada 83 code.
Reference: Chapter 3, page 29
- Fund maintenance contracts on existing tools to enable the project to upgrade to the Ada 95 versions as they become available.
Reference: Chapter 3, page 30
- Review early adoption lessons learned reports to minimize risk.
Reference: Chapter 3, page 19
- Matrix, contract, or train Ada 95 expertise for the Program Office to support technical performance monitoring and evaluation at milestone reviews or technical interchanges.
Reference: Chapter 3, page 41