



Functionality

- 1) Overall, does the product feel intuitive?
- 2) Will it support future development activities as well as your development today?
- 3) Does it support all SQL Types including packages? Does it recognize the creation order needed for complex relationships of views, services programs, and types, and provide for easy management?
- 4) How does the product handle exceptions to the norm? Is it easy to override or customize?
- 5) Does it manage cross platform deployments?
- 6) How does it handle approval processing and tracking?
- 7) Does it provide source compare and conflict management functionality?
- 8) Does the product appear easy to administer?
- 9) Is audit reporting part of the product, or does it have to be purchased separately?
- 10) Can source be checked out via the IFS or imported from a source repository such as Git or SVN?
- 11) Are you required to learn a new language to use the tools?

Project Management

- 12) Can approval be done by business users without direct access the CM product?
- 13) Discuss your approval and notification requirements to determine if the product will meet them.
- 14) If you use another project/service desk tool, is there currently an interface to the CM product? In most cases, this is done via RESTful APIs.

IBM i Specific Check Points

- 15) Are there easy ways to manage queries, menus, control tables, and SQL?
- 16) Rollback capabilities are not all created equal; make sure you know what can and can't be done.
- 17) What are the limits on the number of test levels between development and production?
- 18) Does it support pre/post install commands per object and per deployment?
- 19) What type of cross-referencing is available within the product: Field level or Object level?
- 20) Can you link native and non-native deployments?
- 21) Can source be compiled on target systems?
- 22) Can source be stored in a Git or SVN repository?

IBM i Specific Check Points - Eclipse

- 23) If you use RDi, does it support your version?
- 24) Can all programmer functions be managed via the plugin?
- 25) Are there consistent plugin tools for native and non-native support?





Software Vendor Interfaces

- 26) Do vendor interfaces incur additional licensing cost?
- 27) Who wrote the interface and how is it supported? Many older 3rd party interfaces have minimal or no support.
- 28) Does the CM product provide RESTful APIs to support access?
- 29) Is the vendor or a 3rd party responsible for bug fixes and upgrades for the interface?

About the Vendor

- 30) How long has the software provider been in business?
- 31) How often are new releases generated? This indicates the vendor's commitment to the product.
- 32) Take a look at the vendor's major release notes. Are they providing new functionality or just bug fixes?
- 33) Ask for references that use the product in much the same way as you will. Call them and ask lots of questions:
 - How did the implementation go? Request timeline.
 - How well do interfaces function?
 - How often do they have to contact Tech Support? How quickly are problems resolved?
 - How does the vendor respond to enhancement requests?
 - If they migrated from another CM tool, how smooth was the transition? What historical information were they able to keep? How did work in progress move into the new product?
- 34) Is the vendor owned by the original investors or by a technology holding company? Investor groups and holding companies provide financing, but keep in mind that R&D is usually low priority.

Pricing and Availability

- 35) Does the vendor offer a proof of concept? Is there a cost?
- 36) Are their prices in line with competition?
- 37) Is the vendor willing to discuss your environment and provide out-of-the-box solutions?
- 38) What are their training rates and options?
- 39) Must additional modules be purchased to support your environment? Evaluate total cost.
- 40) Bells and Whistles – Do you need all the 'extras' they are proposing?
- 41) What tools are incorporated into the product? Can this eliminate tools you are currently paying extra for?
- 42) Are there additional license costs for High Availability and Disaster Recovery systems?

The Software License

- 43) Does the license protect your investment with a money-back guarantee period?
- 44) Warranties should not include blanket "software provider is not responsible if nothing works" statements.
- 45) The license should not allow for any post-sale changes of licensing or maintenance fees.
- 46) Maintenance fees should be specified, usually as a %. Is maintenance based on actual purchase price or retail?
- 47) Watch for kill switches or "back doors" programmed into the application for purposes of disabling the software.
- 48) Are you purchasing or leasing the software? Must you pay maintenance for the product to continue to work?

