Deliverables self-checklist for programmers

Version 1.0

March 26, 2019

Deliverables self-checklist for programmers

No	Broad categor	Specific category	Check point	Details of check	Nablarch dependent	Note	Check for each process architecture			Checked by	Date	Comment
							Screen	Rest	Batch			
1	All	General	Line break code in files	- The line break code in files must be the code specified for the project.		* Example Files such as Java, SQL, JSP, Shell and config files must all be checked in the same way.	X	X	X			
2	All	General	Character code in files	- The character code in files must be the code specified for the project.		* Example Files such as Java, SQL, JSP, Shell and config files must all be checked in the same way.	X	X	X			
3	Java	General	Judgment of "" or Null	- The following API provided by Nablarch must be used for "" or Null judgments. - Check whether a value is set ->StringUtil#hasValue - Check whether no value is set ->StringUtil#isNullOrEmpty - Unnecessary logical operators must not be used. Example of unacceptable operator: if(!StringUtil.isNullOrEmpty(hoge)) Use StringUtil#hasValue in this case.	X	* When checking whether a value is set Incorrect: if (!"".equal(value)) {} if (value != null) {} if (!StringUtil.isNullOrEmpty(value)) {} Correct: if (StringUtil.hasValue()) {} * When checking whether a value is not set Incorrect: if ("".equal(value)) {} if (value == null) {} if (!StringUtil.hasValue(value)) {} Correct: if (StringUtil.isNullOrEmpty(value)) {}	X	X	X			
4	Java	General	Joining of character strings	- The character string "null" cannot be used to join character strings that contain a null character.		* How to address this To join character strings that may be null, use a method such as (Apache Commons) StringUtils#join.	X	X	X			

5	Java	General	Close process during reading and writing of files	- During reading and writing of files, refer to the points below to check whether there is a close process. (1) When writing to HttpResponse using DataRecordResponse ->Not needed as a close process is performed on the framework side. (2) When writing a file using FileRecordWriterHolder ->Not needed as a close process is performed on the framework side. ->However, FileRecordWriterHolder can only be used during batch processing (it cannot be used during screen processing). (3) When reading and writing files using DataRecordFormatter ->A close process must be executed. (4) When reading and writing files using a different class (standard Java class) ->A close process must be executed.	X		X	X	X			
6	Java	General	Method arguments	- Unused arguments must not be defined. - The minimum necessary arguments for the method must be defined.		* Example: If only one property in Dto is used, use only the property (not Dto) as an argument.	X	X	X			
7	Java	General	Boolean method arguments	- Boolean must be passed to the method arguments, and branching must not be used when processing these values.		* Reason In many cases, it is best to branch at the origin of the call and call separate methods.	X	X	X			
8	Java	General	Magic numbers	- Magic numbers must be changed to constants and given suitable variable names.			X	X	X			
9	Java	General	Variable names that are too abstract	- Variable names that are too abstract must not be used.		* Variable names that are too abstract Example of an unacceptable name: Pattern PATTERN This variable name does not indicate the purpose of the pattern. The intent needs to be clear from the variable name. Example of a correction: EXTRACT_HOSTNAME_PATTERN	X	X	X			
10	Java	General	Type-specific variable and method names	- Type-specific names must not be used for variable names or method names.		* Examples Examples of unacceptable names: XXXMap, XXXStr, etc.	X	X	X			
11	Java	General	Names of methods returning Boolean	- Names such as isXXX, hasXXX or canXXX must be used. - A name that clearly indicates the return value (true or false) must be used in the area indicated by XXX.		* Example Example of an acceptable name: hasError ->It can be envisioned that true will be returned in the event of an error.	X	X	X			
12	Java	General	Character string comparison with constants	- Constant.equals(hoge) must be used (not hoge.equals(constant)).		* Reason hoge.equals(constant) may cause a NullPointerException.	X	X	X			

13	Java	General	Method for judging zero items	- Collection#isEmpty() must be used for this judgment when handling collections.		* Reason The code will be too long if zero items are judged after acquiring the size.	X	X	X			
14	Java	General	Dates	- The date format set to be used must not be incorrect.		* Examples Business date, system dates, etc.	X	X	X			
15	Java	General	Nest for method calls	- Up to two method call nests can be used.		* Reason Readability decreases if too many method calls are included in a method argument. Example of unacceptable code: setZZZ(convertXXX(getXXX(hoge,huga),getYYY(moge,moga).toString().subString(0,12))); Example of acceptable code: String foo = getXXX(hoge,huga); String bar = getYYY(moge,moga).toString().subString(0,12); setZZZ(convertXXX(foo, bar));	X	X	X			
16	Java	General	Sorting order of methods	- The order "public, protected, private" must be used.			X	X	X			
17	Java	Action	Error destination	- A forwarding destination must be set correctly for errors such as exclusive errors and validation errors.			X					
18	Java	Action	Automatically set items	- Values that are set automatically, such as @CurrentDateTime, cannot be set with individual business logic.	X		X	X	X			
19	Java	Action	Instance variables	- Actions must not have instance variables.	X	* Exception Cases in which functions such as the counter of batch output items need to be achieved using an instance variable.	X	X	X			
20	Java	Form	Domain definitions for forms	- Domain definitions specified in the specifications must be checked and the close investigation process defined in the domain definitions must be correctly implemented.			X	X	X			
21	Java	Database access	Exclusive control	- Exclusive control using UniversalDao must be implemented for update processes.	X	* Implementation method 1. Store the records to be updated in an entity. *For screens, the above entity is stored in a session store when displayed for the first time and is held for the duration of the request. 2. Overwrite the items to be updated in the entity. *Version numbers do not need to be overwritten in the entity as these are updated automatically by the framework. 3. Execute the update process using UniversalDao.update(entity).	X	X				

22	Java	Database access	Database access	- As a general rule, UniversalDao must be used for database access. - A batch execution method (UniversalDao#batchInsert, etc.) must be used if the same SQL syntax is issued multiple times.	X	* Operations provided by UniversalDao - Registration/batch registration - Registration/batch registration with a primary key specified *Used only for updating under exclusive control - Deletion/batch deletion with a primary key specified - Searching with a main key specified - Searching using SQL syntax	X	X	X			
23	Java	Database access	Multiple search results	- If SQL enabling multiple search results to be acquired is issues, the number of results must be judged correctly. - If one result is sufficient, SQL must be implemented so that only one result can be acquired. - SqlResultSet#get(0) must not be executed for SELECT expressions for which multiple results are returned.	X	* Reason If SqlResultSet#get(0) is executed for SELECT expressions that return multiple results, this may be due to issues such as missing search condition specifications or insufficient considerations during the design stage. Check the specifications.	X	X	X			
24	Java	Database access	Zero search results	- Implementation when there are zero search results must be taken into account. Check the specifications and confirm that one of the following methods is used. (1) If the specifications indicate that zero results are possible - Zero results must be handled as indicated in the design specifications (e.g. error handling consisting of an on-screen user notification) (2) If the specifications indicate that zero results are not possible - Handling of zero results must not be implemented.	X	* Examples of handling of zero search results - Displaying "No search results" on screen - Raising an application exception	X	X	X			
25	Java	Database access	Method for acquiring number of search results in list search	- UniversalDao#countBySqlFile must be used. - SqlResultSet#isEmpty must not be used, as this method judges the number of results on the current page.	X		X	X	X			
26	Java	Database access	Acquisition of values from SqlRow	- A suitable acquisition method must be used (not casting with SqlRow#get()).	X	* Examples SqlRow#getString(), SqlRow#getBigDecimal(), etc.	X	X	X			
27	JSP	General	Button submission control	- type="button" must be specified when using the following custom tags so that submission is not executed when the enter key is pressed. - n:button - n:popupButton - n:downloadButton	X	* Examples type="button" specified as a tag attribute. Example of unacceptable code: <n:button uri="hoge/000">fuga</n:button> Example of acceptable code: <n:button type="button" uri="hoge/000">fuga</n:button>	X					
28	JSP	General	Linking of popup windows for input items	- n:changeParamName must be used when changing the request parameter name of information input in a screen and sending the information to a popup window.	X	* Reason When the request parameter name is changed using n:param before being sent, the value that is sent is the value indicated when the screen is displayed for the first time.	X					

29	JSP	General	Submission outside the application	- n:a must be used (not n:submit) for submissions outside applications.	X	* Reason The n:a tag needs to be used because submissions outside the application do not have a request ID and are all inactive when used as "submit" links. * Exceptions - When linking values to the main screen from a popup window - The close button - Links in the page	X					
30	JSP	General	Screens with only one input item	- For screens with only one input item with an <input type="text"> format (<n:text>, etc.), this must be placed above the text box containing dummy text (display:none & disable) as an error occurs if Enter is pressed with the focus on the text box.	X	* Examples - Screens where there is only a search condition (search box) and a search button - Screens where there is only search condition 1 (radio button), search condition 2 (text box) and a search button *This is not dependent on the version of Nablarch. It is caused by the browser.	X					
31	Testing	General	Assertion of expected values	- A suitable assertion method must be called for assertion targets.		* Examples - assertEquals type - assertSqlRowEquals - assertSqlResultSetEquals	X	X	X			
32	Testing	General	Table assertion	- EXPECTED_COMPLETE_TABLE must be used for table assertion.	X	* Reason Omitted columns are not targets for comparisons when EXPECTED_TABLE is used, but are compared as columns containing default values when EXPECTED_COMPLETE_TABLE is used.	X	X	X			
33	Testing	General	Exception testing	- @Test(expected=IllegalArgumentException.class) must be used (sending of exceptions must not be confirmed using try-catch). However, ExpectedException must be used when confirming the details of exceptions. - If multiple types of exception occur for one method, a separate test method must be used for each type of exception.	X		X	X	X			
34	Testing	General	Reviewed content	- Content to be reviewed must be checked to confirm that there are no omissions.		* Reason The following files are easily omitted, so care needs to be taken. - SQL files - XML files - fmt files	X	X	X			

X: Dependency