Honza, you are right. A named object has to be std:move()'ed to communicate its moveable (temporary) status to the compiler, even when it's declared as temporary (using &&). std::move will appear a lot in new library code and user code as well, so people will get use to it soon.

You can do in move assignments or move constructors the same thing you used to do in pick constructor and pick assignment, eg, mark the right hand object as picked.

Now we can do

Vector<int> b=std:move(anotherVectorIntInstance); // move construction;

//or
Vector<int> b=anotherVectorIntInstance; //copy construction or deep copy, instead of Vector<int> b(anotherVectorIntInstance, 0); // using dummy param to signal the intention of deep copy.

provide the underlying U++ facilities are modified using the rvalue reference language feature.